

# Введение в Web-технологии. Структура HTML

Модуль 1 (2 пары)

## Что такое HTML.

- **HTML** (от англ. *HyperText Markup Language* — «язык гипертекстовой разметки») — является на данный момент стандартом для документов, передаваемых в Интернете. HTML-разметка интерпретируется браузерами в виде отформатированного текста и может отображаться на экране монитора компьютера, планшета или смартфона.
- HTML базируется на основе **SGML** (*Standard Generalized Markup Language* — стандартного обобщённого языка разметки), который появился в 1986 г., и соответствует международному стандарту ISO 8879. Этот язык был изначально предназначен для структурной разметки текста, но не содержал описания внешнего вида документа, который на нем можно было бы создать.
- SGML подразумевал описание синтаксиса для написания главных элементов разметки текстов, причем уже тогда они получили название «**тегов**», он сам по себе не являлся системой для разметки текста и не имел списка структурных элементов языка для того, чтобы использовать их при создании документа.

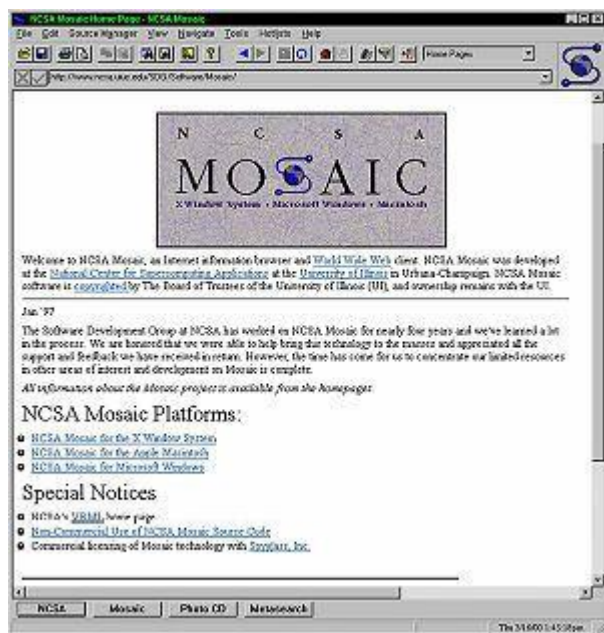
# История создания



В 1991 году Европейский институт физики частиц (CERN) в Женеве, Швейцария, объявил о необходимости разработки механизма, позволяющего передавать гипертекстовую информацию через Глобальную сеть. За разработку этого языка взялся сотрудник этого института Тим Бернес-Ли. И именно стандарт SGML лег в основу будущего языка **HTML** — *Hyper Text Markup Language*. В соответствии с требованиями и нуждами CERN Тим Бернес-Ли разрабатывал HTML в первую очередь для обмена научной и технической документацией.

- Был определен небольшой набор структурных и семантических элементов — дескрипторов, которые затем стали называть тегами, которые позволяли создавать достаточно простые и в тоже время красиво оформленные документы. Таким образом, HTML успешно решал проблемы, оставшиеся от SGML. Работая в CERN, Тим Бернерс-Ли занимался не только развитием HTML. В его задачи входило построение внутренней сети организации. Концепции, реализованные в ней, были доработаны и переросли в проект под названием «Всемирная паутина». Проект подразумевал, что можно будет публиковать документы, размеченные при помощи HTML, в открытом доступе. Все документы должны иметь. Для реализации своей идеи Бернерс-Ли создал специальные программы: HTTP-сервер и WEB-браузер. Первый в мире веб-сайт был размещён 6 августа 1991 года по адресу <http://info.cern.ch/>. В его содержании описывается принцип работы сети, как установить веб-сервер и создать простую страницу.

# Стандарты HTML



В 1993 г. появился HTML 1.2, который содержал 40 тегов. Эти теги полноценного оформления страницы еще не давали. Для отображения HTML-страниц была разработана специальная программа - «браузер». И первым браузером стал «Mosaic», который был разработан в Национальном центре суперкомпьютерных приложений США (NCSA). За первый год было установлено около двух миллионов копий этой программы. Она поддерживала отображение картинок, распространялась бесплатно и помещалась на одну дискету.

Следующим этапом в развитии уже стандарта HTML было создание в апреле 1994 года Консорциума W3C (*World Wide Web Consortium*).

Поскольку официальной спецификации **HTML 1.0** не существовало, именно W3C начал заниматься подготовкой спецификации HTML следующей версии. Ей сразу присвоили номер 2.0, чтобы подчеркнуть отличие от других версий HTML. Первый результат удалось получить спустя год насыщенной работы — в 1995 году. Из больших дополнений нужно отметить создание механизма форм для отсылки информации с компьютера пользователя на сервер.

# Стандарты HTML

Параллельно с разработкой **HTML 2.0** велась разработка **3-ей версии** стандарта, которая появилась в марте 1995 года и содержала в себе теги для создания:

- математических формул;
- страниц;
- примечаний;
- вставку рисунков, обтекаемых текстом;
- поддержку формата gif и т.д.

При том, что этот стандарт был совместим со второй версией, реализация его была сложна для браузеров того времени.

Кроме того, в 1995 году уже существовала необходимость в большем визуальном разнообразии страниц. Средств, предлагаемых HTML, особенно в рамках стандарта SGML, было недостаточно. Тогда корпорация W3C приступила к созданию дополнительной системы, не противоречащей основам HTML, но при этом позволяющей описывать визуальное оформление документов. В результате появился **CSS** — *Cascading Style Sheets* — Каскадные таблицы стилей, которые представляли собой иерархические стилевые спецификации. У них был собственный синтаксис, структура и задачи, которые позволяли дополнить теги HTML визуальным форматированием. Создание CSS было большим шагом вперед, т.к. потребности в визуальном представлении страниц сильно возросли, а HTML не предлагал для этого никаких средств.

# Стандарты HTML

- Версия HTML 3.1 официально никогда не предлагалась, поэтому следующей версией стандарта **HTML стала 3.2**, выпущенная *14 января 1997 года*. В ней были опущены многие нововведения из версии 3.0 и добавлены нестандартные элементы, поддерживаемые браузерами Netscape Navigator и Mosaic.
- *18 декабря 1997 года* была принята четвертая версия HTML. **Стандарт HTML 4.0** содержал, как и в третьей версии, много элементов, специфичных для отдельных браузеров. В 4-ой версии многие элементы были помечены как *устаревшие и не рекомендуемые к использованию*. Вместо них рекомендовалось использовать таблицы стилей CSS.
- HTML 4.01, который был утвержден 24 декабря 1999 года. В нем были приняты значительные изменения, и этот стандарт был популярен в сети довольно долго.
- Параллельно с разработкой стандарта HTML ведется также разработка альтернативного стандарта **XHTML** (*Extensible Hypertext Markup Language*) — расширяемый язык разметки гипертекста, который использует подход XML. Вариант XHTML 1.1 одобрен в качестве рекомендации консорциума *31 мая 2001 года*. Фактически разработка новых стандартов HTML никогда не прекращалась, т.к. требования к отображению страниц постоянно увеличиваются.
- С 2007 года разрабатывался стандарт HTML5, который был окончательно утвержден *28 октября 2014 года*. А с 17 декабря 2012 года уже ведется разработка стандарта HTML 5.1.

# Война браузеров

С 1994 года в сети появился **Netscape Navigator** и **Opera**, а с 1995 года — **Internet Explorer**. Чтобы отхватить как можно больший кусок потенциальной аудитории, привлечь максимальное количество новых пользователей, **Netscape** вводила в **HTML** все новые и новые усовершенствования, которые поддерживались, естественно, только одноименным браузером. Практически все новые теги были направлены на улучшение внешнего вида документа и расширение возможностей его форматирования.

Летом **1996** г. была выпущена версия Internet Explorer 3.0. Она поддерживала практически все расширения Netscape и обладала симпатичным и дружелюбным интерфейсом. Поэтому очень быстро утвердилась в качестве «второго главного браузера».

Четвертые версии обоих браузеров вышли практически одновременно и не отличались друг от друга особой скоростью работы или иными параметрами. Если Netscape нужно было покупать, то Internet Explorer начал поставляться бесплатно в операционной системе Windows и стал фактически стандартом отрасли. 90-е годы 20-го века ознаменовались таким понятием как «Война браузеров», которое, по сути, обозначает противостояние на рынке Интернета Netscape Navigator и Internet Explorer. Проблемой эта война стала для верстальщиков, т.к. каждый браузер старался внести свою лепту в развитие языка HTML и не слишком-то следовал стандартам W3C. Поэтому нередко на страницах сайтов можно было найти запись «Корректно отображается в браузере...» со ссылкой на скачивание соответствующей программы.



# Война браузеров

- В конце 1990-х — начале 2000-х годов за счет того, что IE входил в Windows по умолчанию, он стал самым популярным, особенно для не опытных пользователей.
- Из NetscapeNavigator со временем родился Mozilla FireFox, который поддерживал стандарты W3C, имел вкладки, а не окна и ряд фишек, переманивших на свою сторону продвинутых пользователей и верстальщиков.
- Достаточно много почитателей появилось и у браузера Opera. Следующим новым браузером стал Google Chrome, который был выпущен в свет 11 декабря 2008 года, и его исходный код стал доступен под названием Chromium. С этого момента любая компания могла сделать свой браузер на основе этого кода. Например, так появились Yandex-браузер или Amigo. Кроме того, Opera тоже перешла на движок WebKit, на котором был создан Chrome.
- Google Chrome предоставляет удобные инструменты для разработчиков исходного кода — html-верстальщиков, JavaScript-программистов.
- *Инструменты разработчика* сейчас присутствуют в любом браузере, в том числе и в Internet Explorer последних версий. И не последнюю роль здесь сыграло дополнение для FireFox с названием FireBug, которое предоставляло информацию об html-элементах, css-правилах для них и позволяло отлаживать код JavaScript. В Opera для тех же целей существовало расширение Dragonfly.
- **«Кроссбраузерность»** - подразумевает одинаковое отображение сайта во всех браузерах. На данный момент эта проблема стоит уже не так остро, как в 2005–2012 гг., т.к. все современные браузеры стараются поддерживать стандарты организации W3C. Тем не менее, сверстав сайт, следует проверить, как он отображается в самых популярных на данный момент браузерах.



# Война браузеров

<b>2017</b>	<b><u>Chrome</u></b>	<b><u>IE/Edge</u></b>	<b><u>Firefox</u></b>	<b><u>Safari</u></b>	<b><u>Opera</u></b>
July	76.7 %	4.2 %	13.3 %	3.0 %	1.2 %
June	76.3 %	4.6 %	13.3 %	3.3 %	1.2 %
May	75.8 %	4.6 %	13.6 %	3.4 %	1.1 %
April	75.7 %	4.6 %	13.6 %	3.7 %	1.1 %
March	75.1 %	4.8 %	14.1 %	3.6 %	1.0 %
February	74.1 %	4.8 %	15.0 %	3.6 %	1.0 %
January	73.7 %	4.9 %	15.4 %	3.6 %	1.0 %

<b>2016</b>	<b>Chrome</b>	<b>IE/Edge</b>	<b>Firefox</b>	<b>Safari</b>	<b>Opera</b>
December	73.7 %	4.8 %	15.5 %	3.5 %	1.1 %
November	73.8 %	5.2 %	15.3 %	3.5 %	1.1 %
October	73.0 %	5.2 %	15.7 %	3.6 %	1.1 %
September	72.5 %	5.3 %	16.3 %	3.5 %	1.0 %
August	72.4 %	5.2 %	16.8 %	3.2 %	1.1 %

## Базовые определения HTML

- Создание HTML-документа предполагает разметку всего его содержимого в виде тегов (первое название — дескрипторов), которые записываются в угловых скобках. Текст должен располагаться между *открывающим (начальным)* и *закрывающим (конечным)* тегами.

```
<p> Текст абзаца </p>
```

- Весь текст, заключённый между начальным и конечным тегом, включая и сами эти теги, называется **элементом**. Сам же текст между тегами является **содержанием элемента**. Содержание элемента может включать в себя любой текст, в том числе и другие элементы. Для любого тега можно задать дополнительные свойства, которые называются **атрибутами**. Размещаются они только в открывающем теге и представляют собой пары вида «свойство="значение"», разделенные пробелами.

```
<p id="test" class="par"> Текст абзаца </p>
```

- HTML является регистронезависимым языком разметки, поэтому допустимо писать теги как в верхнем, так и в нижнем регистре. Тем не менее, правилом дефакто стало уже давно написание тегов только **в нижнем регистре**. Документы HTML, по сути, являются текстовыми документами. Они имеют расширение **.html** или **.htm**, и для их просмотра необходим браузер, а для изменения — редактор кода.

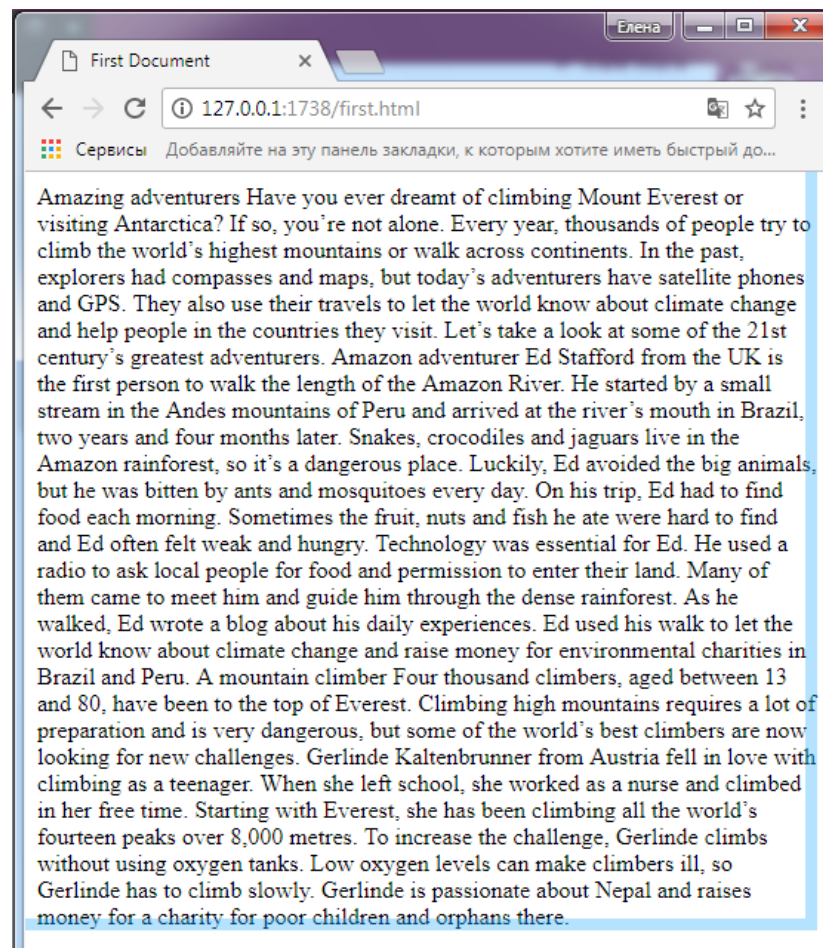
# Базовые определения HTML

- Еще одной особенностью HTML является **игнорирование пробелов** и переносов строк. Как бы ни был отформатирован текст в вашем html-документе *ДО добавления тегов* — с отступами, сделанными клавишей TAB, с переносом каждой строки (клавиша ENTER) — в браузере он будет просто сплошным текстом

```

1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>First Document</title>
6  </head>
7  <body>
8      Amazing adventurers
9
10     Have you ever dreamt of climbing Mount
        Everest or visiting Antarctica? If so,
        you're not alone. Every year, thousands
        of people try to climb the world's
        highest mountains or walk across
        continents. In the past, explorers had
        compasses and maps, but today's
        adventurers have satellite phones and
        GPS. They also use their travels to let
        the world know about climate change and
        help people in the countries they visit.
        Let's take a look at some of the 21st
        century's greatest adventurers.
11
12     Amazon adventurer
13
14     Ed Stafford from the UK is the first person

```



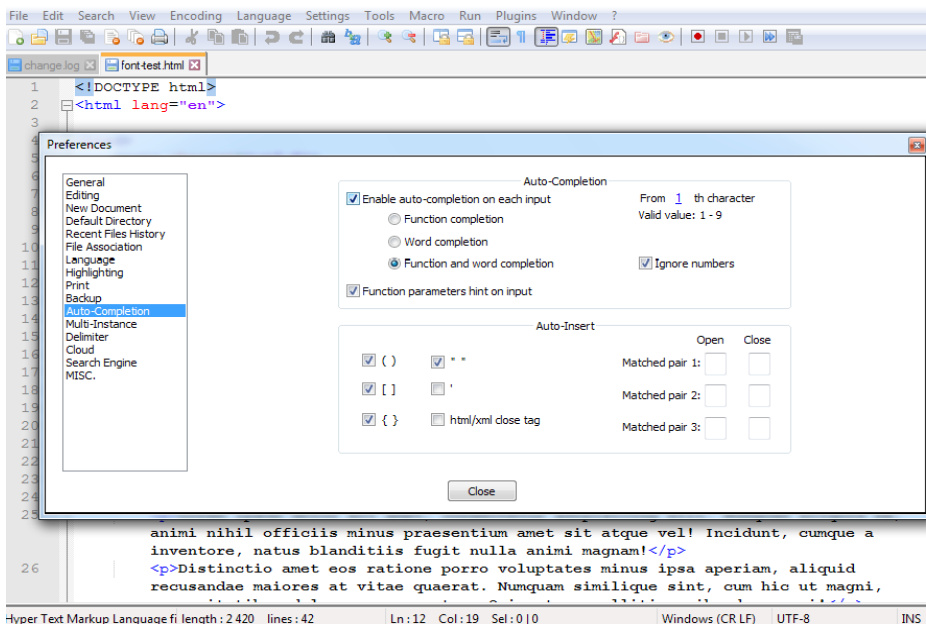
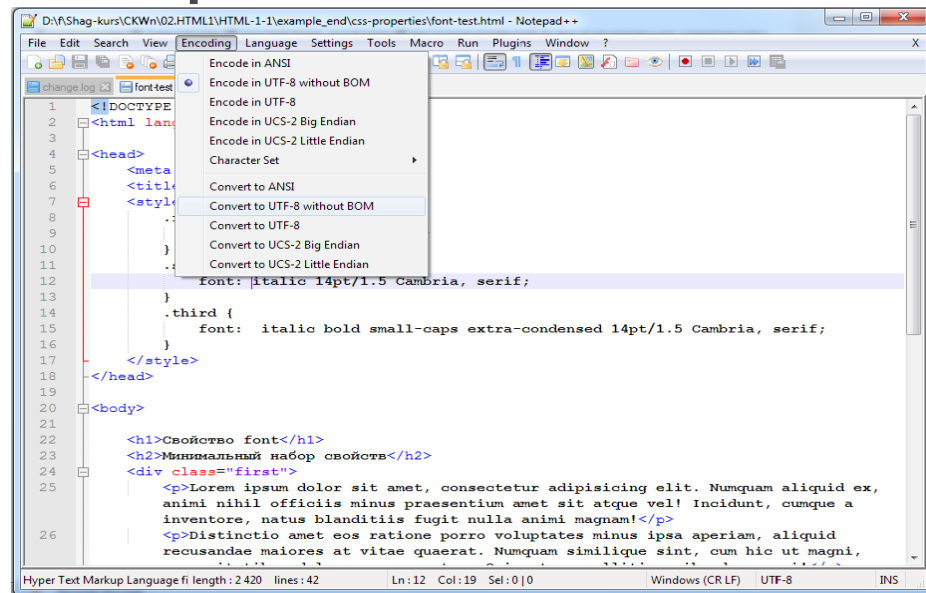
# Базовые определения HTML

- Поэтому весь текст внутри html-документа должен быть разбит на теги соответственно *семантике*, или *внутренней логике* этого документа. И для различных уровней вложенности элементов, которые наверняка будут присутствовать на ваших страницах, **следует делать отступы** для визуального отделения родительских и дочерних элементов

```
<div class="modal fade" tabindex="-1" role="dialog" id="cartDetails">
  <div class="modal-dialog modal-lg">
    <div class="modal-content">
      <div class="modal-header">
        <button type="button" class="close" data-dismiss="modal" aria-
          label="Закрыть"><span aria-hidden="true">&times;</span>
        </button>
        <h2 class="text-center">Корзина</h2>
      </div>
      <div class="modal-body">
        <p>Пока товаров в корзине нет</p>
      </div>
      <div class="modal-footer">
        <button type="button" class="btn btn-order" data-dismiss="modal" data-
          text="Продолжить покупки">Продолжить покупки</button>
        <button type="button" id="checkoutBtn" class="btn btn-success" data-
          text="Оформить заказ">Оформить заказ</button>
      </div>
    </div>
  </div>
</div>
```

# Выбор редактора кода

# Notepad++



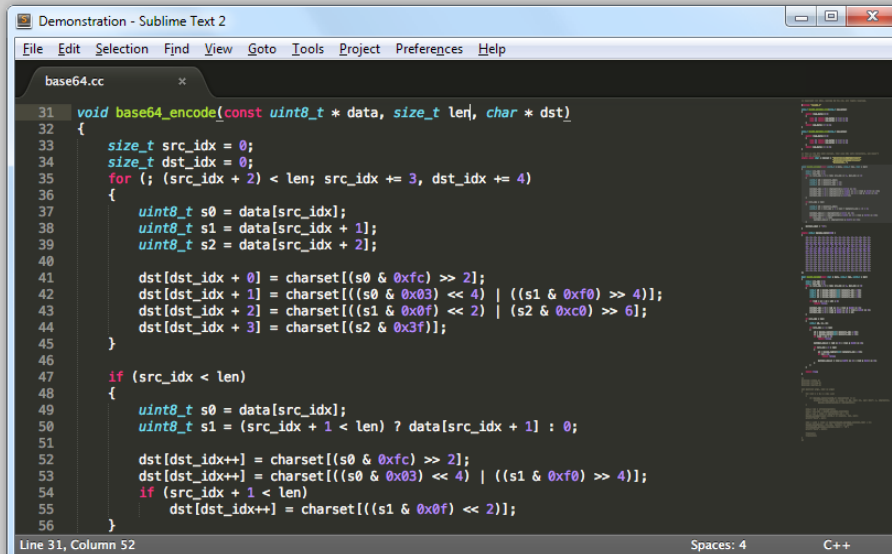
- Бесплатный текстовый редактор, который поддерживает и подсвечивает синтаксис более 100 языков. Может открывать файлы в различных кодировках. На официальном сайте <https://notepad-plus-plus.org/> можно скачать последнюю версию программы. Этот редактор кода удобен, быстро загружается. В нем можно настроить под себя некоторые функции через меню Опции -> Настройки, например, автозаполнение, т.е. закрытие скобок, тегов и т.п. Notepad++ разработан только для систем Windows 32- и 64-bit

# Sublime Text

Home Download Buy Blog Forum Support

## Sublime Text

Sublime Text is a sophisticated text editor for code, markup and prose. You'll love the slick user interface, extraordinary features and amazing performance.



```

31 void base64_encode(const uint8_t * data, size_t len, char * dst)
32 {
33     size_t src_idx = 0;
34     size_t dst_idx = 0;
35     for (; (src_idx + 2) < len; src_idx += 3, dst_idx += 4)
36     {
37         uint8_t s0 = data[src_idx];
38         uint8_t s1 = data[src_idx + 1];
39         uint8_t s2 = data[src_idx + 2];
40
41         dst[dst_idx + 0] = charset[(s0 & 0xfc) >> 2];
42         dst[dst_idx + 1] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
43         dst[dst_idx + 2] = charset[((s1 & 0x0f) << 2) | (s2 & 0xc0) >> 6];
44         dst[dst_idx + 3] = charset[(s2 & 0x3f)];
45     }
46
47     if (src_idx < len)
48     {
49         uint8_t s0 = data[src_idx];
50         uint8_t s1 = (src_idx + 1 < len) ? data[src_idx + 1] : 0;
51
52         dst[dst_idx++] = charset[(s0 & 0xfc) >> 2];
53         dst[dst_idx++] = charset[((s0 & 0x03) << 4) | ((s1 & 0xf0) >> 4)];
54         if (src_idx + 1 < len)
55             dst[dst_idx++] = charset[((s1 & 0x0f) << 2)];
56     }

```

- Пожалуй, самый популярный текстовый редактор. Условно бесплатный. Т.е. вы можете скачать его без оплаты, но через определенное время он будет предлагать вам купить лицензию. Официальный сайт <https://www.sublimetext.com/> на данный момент предлагает 3-ю версию редактора для скачивания.

- Sublime Text позволяет работать со многими форматами. Он легко справляется с большими объемами текстовой информации. Редактор расширяется с помощью установки дополнительных пакетов. Минусом является то, что для настройки Sublime Text нужно потратить порядка 2-3 часов, т.к. вам будет недостаточно функциональности «из коробки».



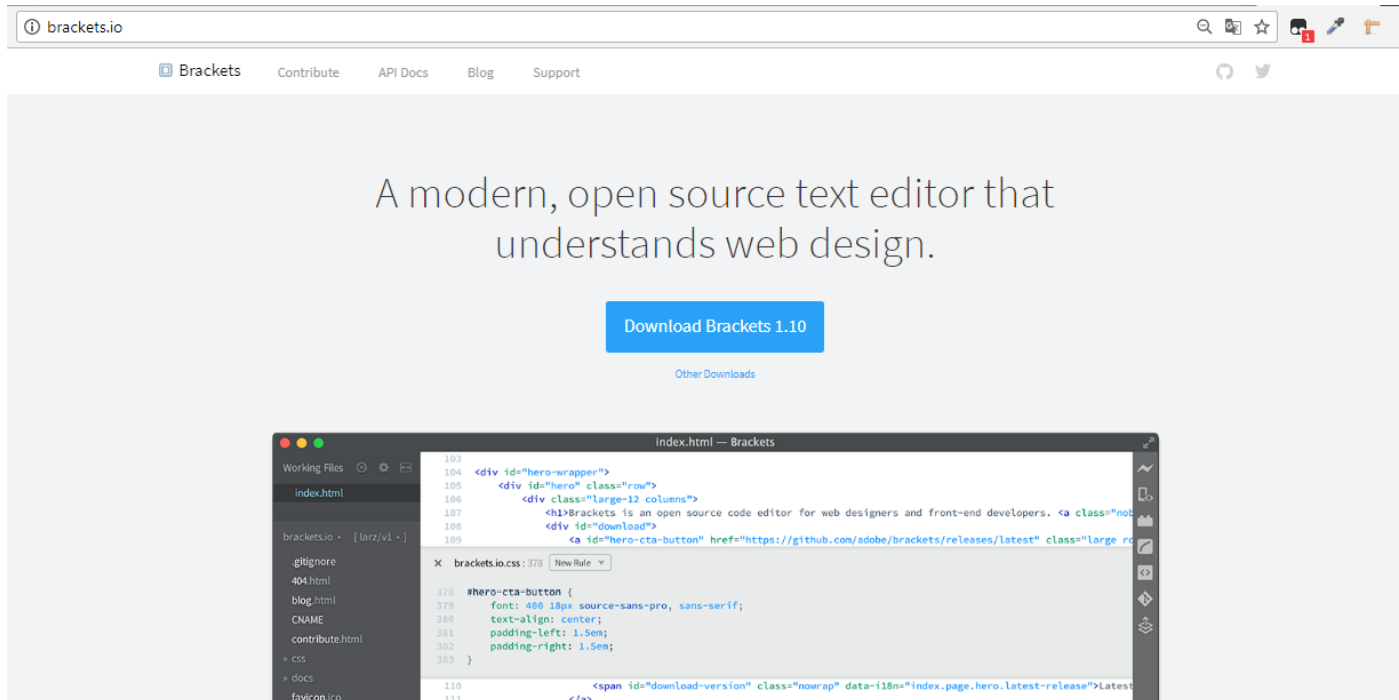
# Atom

- Atom — это текстовый редактор, который является современным, доступным, инструментом, который вы можете настроить для любых действий, но также продуктивно использовать, даже не касаясь файла конфигурации». Этот редактор создавался командой Github. Он бесплатен, имеет открытый исходный код, подходит для macOS, Linux, Windows. Расширяется с помощью плагинов, написанных на Node.js, которые встраиваются под управлением Git Control. Также вы можете устанавливать и менять на нем темы, подбирая цветовые сочетания для подсветки кода, которые будут удобны для ваших глаз.



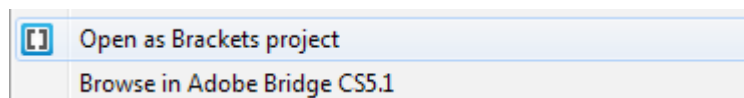
# Brackets

- Бесплатный текстовый редактор, который поддерживает открытие файлов только в формате UTF-8 — самом распространенном на данный момент в web. Разработан он Adobe System для работы в первую очередь с HTML, CSS, JavaScript. Но также позволяет работать с php-файлами и имеет ряд расширений для создания тем для Wordpress, например. Официальный сайт <http://brackets.io/> позволяет скачать последнюю версию редактора. Brackets разработан для систем Windows, MacOS и Linux.
- Расширяется с помощью плагинов, также бесплатных, с открытым кодом, размещенных на Github.

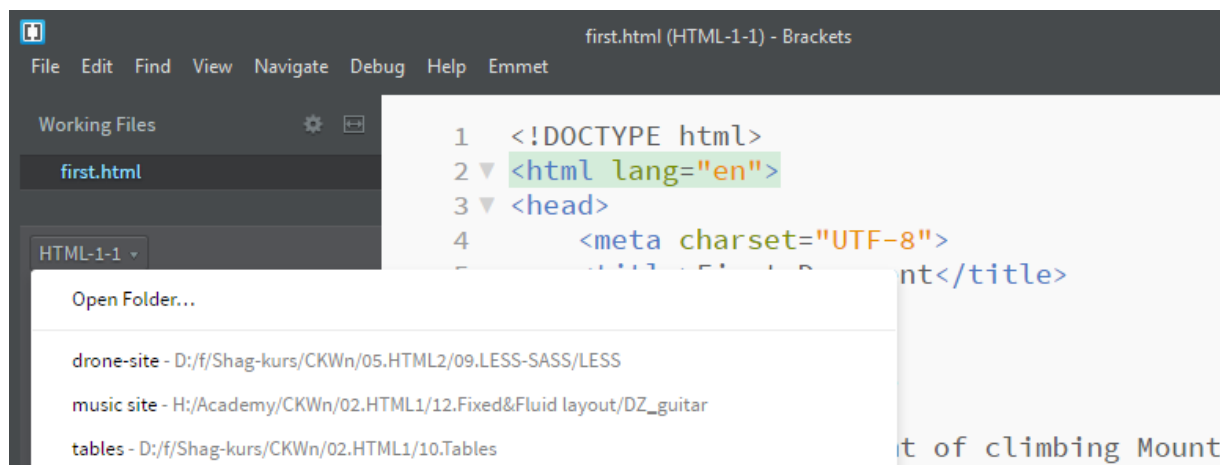


# Использование Brackets для создания и редактирования файлов

- В Brackets лучше работать с проектами — это просто папка на Вашем компьютере или на флешке, которая будет содержать html- и css-файлы, а также папки с изображениями и в дальнейшем с js-скриптами.
- Поэтому создайте у себя на компьютере папку, в которой будете работать и в контекстном меню для этой папки выберите пункт «Open as Brackets Project».

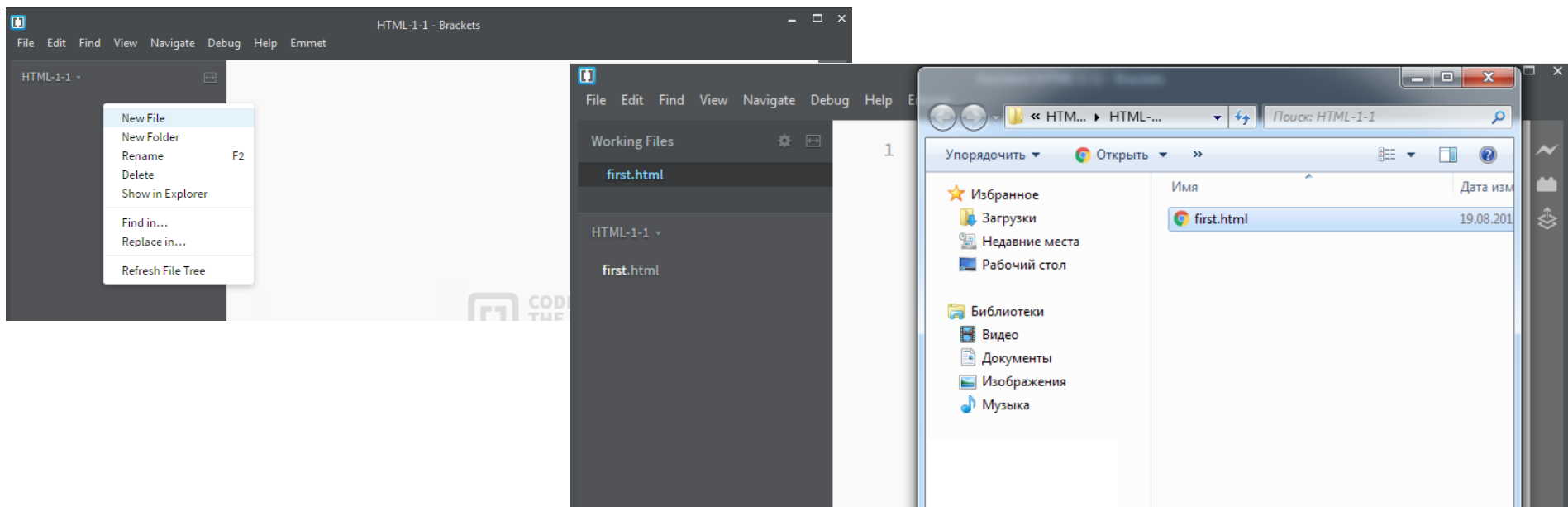


- В этом случае вам будут доступны все файлы и папки внутри проекта. Особенно вы это оцените, когда будете рассматривать тему о ссылках или об изображениях. **Примечание:** если окно *Brackets* уже открыто, вы можете просто перетащить папку с проектом в левую часть редактора или открыть нужную директорию, щелкнув по кнопке со стрелкой.

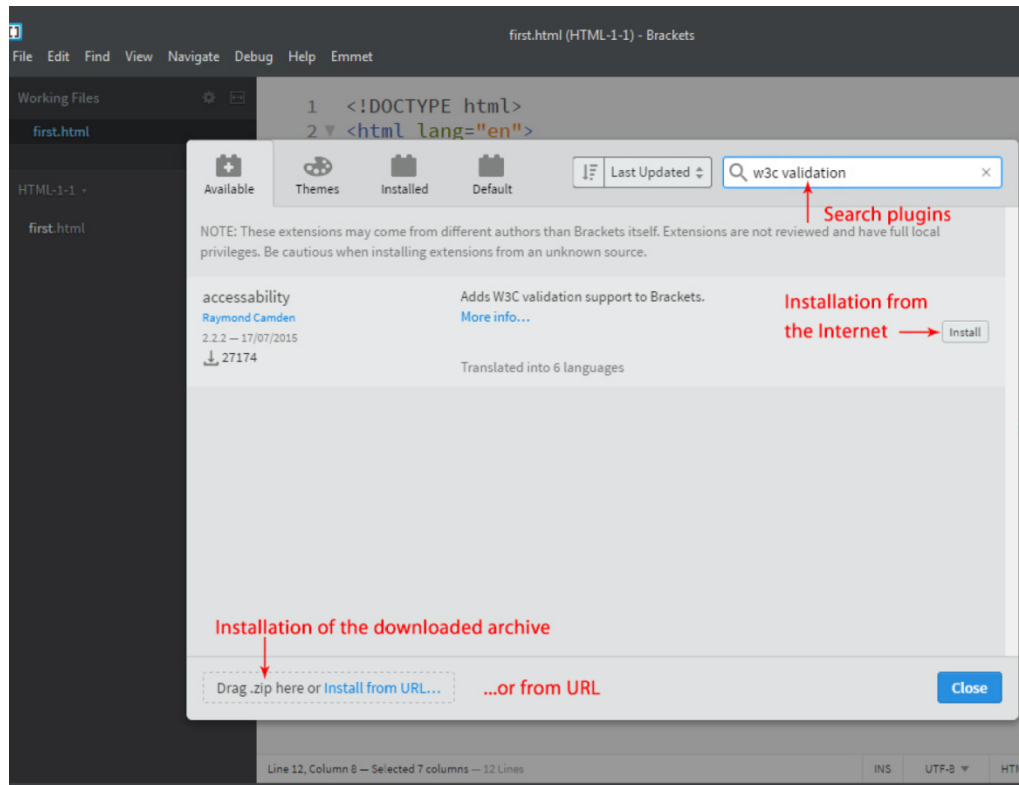


## Создание файлов в Brackets

- Вы можете создавать файлы в формате .html или .htm (а именно этот формат используется в веб) через меню файл->Новый (или Ctrl+N) и потом сохранять файл в нужную папку. Но значительно удобнее создавать файл в левой темно-серой области Brackets по правому клику на ней.
- Веб не любит русскоязычных названий, поэтому все файлы мы будем называть латиницей. В идеале, конечно, нужно это делать на английском языке, но это не является строго обязательным. По умолчанию новый файл будет иметь имя «Без названия-1» и (внимание!!!) не иметь расширения. Поэтому вводим вместо этого first.html — вуаля — файл одновременно создается в нужной папке и доступен для редактирования в Brackets.



# Расширения Brackets



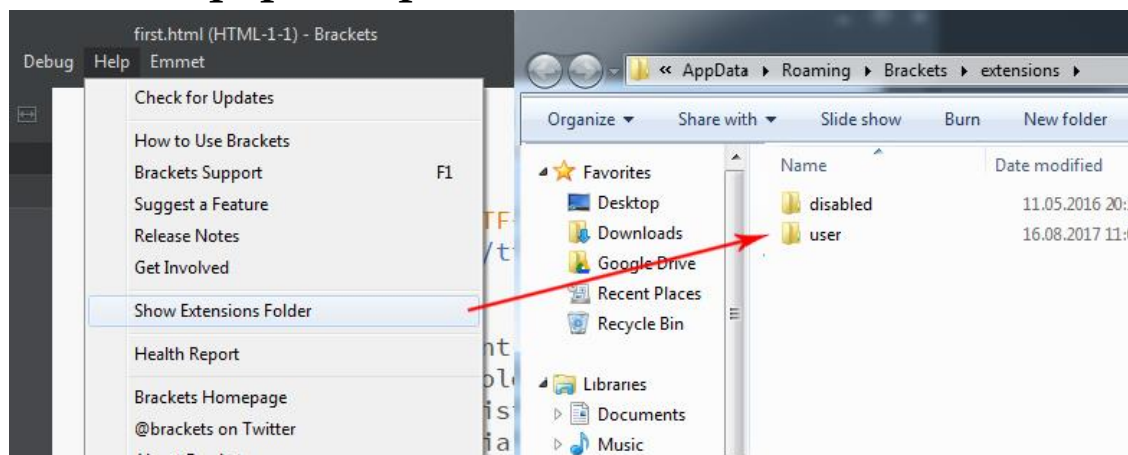
- Плагины, или расширения Brackets, устанавливаются кликом на кнопке в виде лего справа на темно-серой панели Brackets или через меню «Файл -> Менеджер расширений».

- Плагины подгружаются из Интернета на вкладке «Доступные», и их сразу можно устанавливать кликом на соответствующей кнопке. Либо можно найти плагин в репозитории на Github и, нажав на кнопку «Установить с URL», ввести адрес ссылки.

- Плагинов много, поэтому стоит воспользоваться полем поиска. В этом случае из большого списка останутся только те, которые нужны.
- Если соединение с Интернетом отсутствует, а плагины заранее были скачаны в какую-либо папку, то можно установить их, просто перетащив их на кнопку слева внизу с надписью «Drag .zip here».

# Brackets

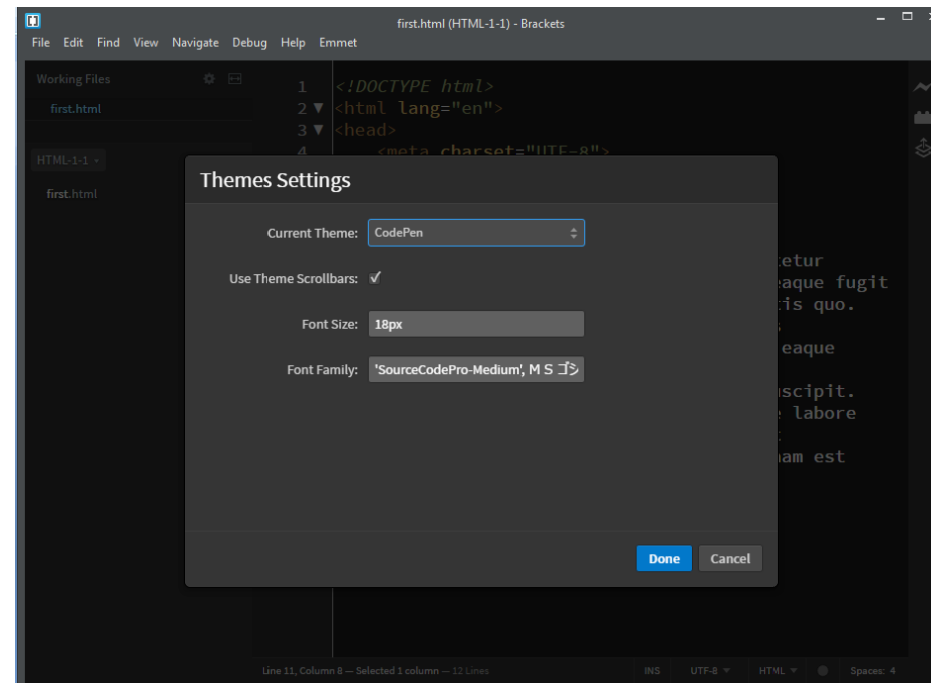
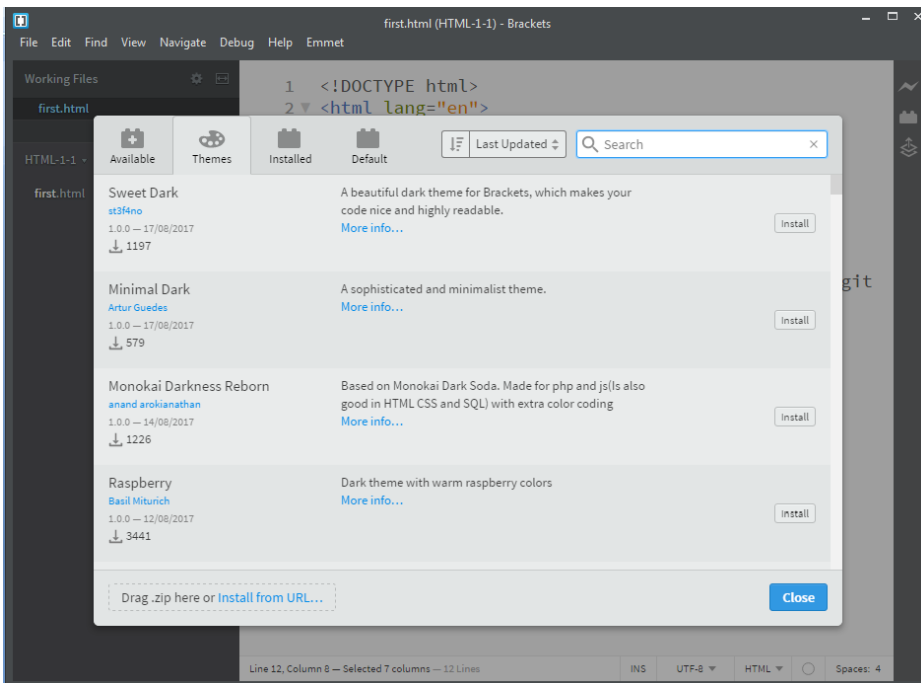
- По поводу рекомендованных плагинов — установите следующий набор:
- 1. Emmet — <https://github.com/emmetio/brackets-emmet>. Этот плагин понадобится для ускорения верстки и написания css-стилей.
- 2. Jsbeautifier — <https://github.com/taichi/bracketsjsbeautifier> или Beautify — <https://github.com/bracketsbeautify/brackets-beautify> - любой из них позволит красиво отформатировать код.



- Если по какой-то причине вы не смогли установить расширение, то можно разархивировать плагин из .zip-файла в папку с расширениями. Отобразить ее можно через меню Помощь->Показать директорию расширений. Откроется папка Roaming->Brackets. В ней будет папка user, в которую и необходимо распаковать архив. После этого **обязательна перезагрузка редактора**

# Темы для Brackets

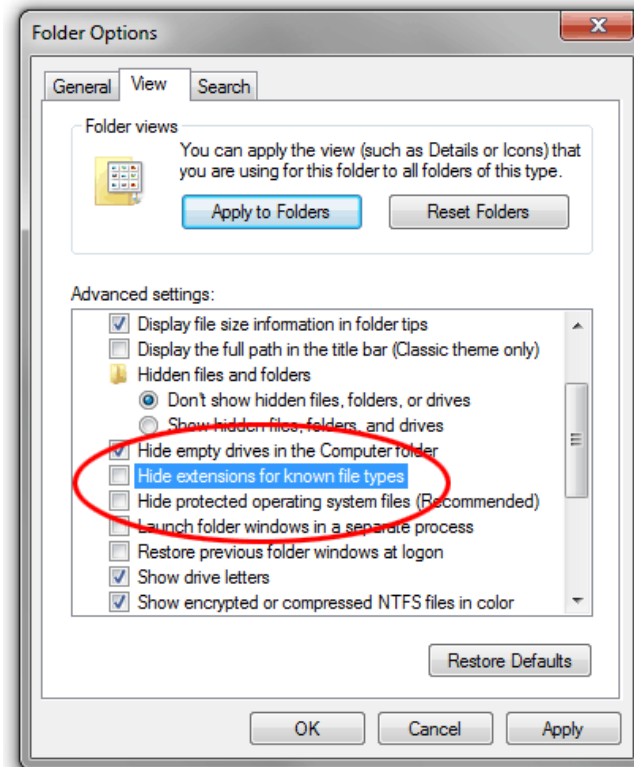
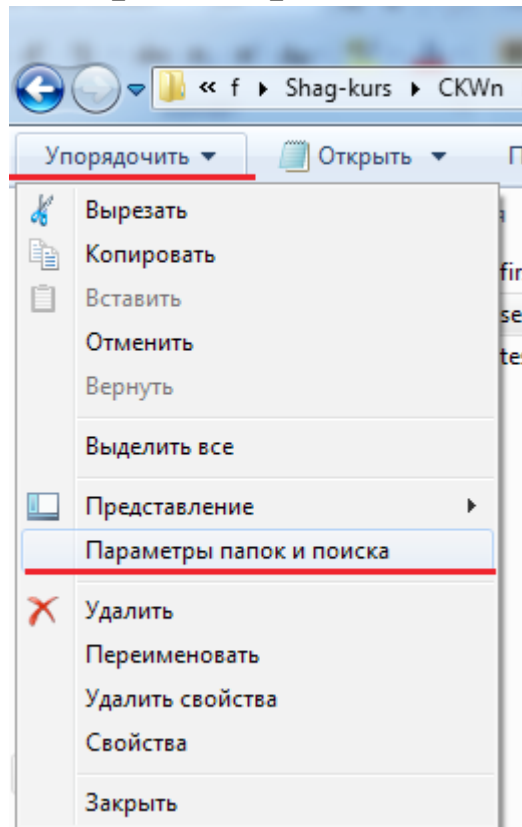
- Загружаются так же, как и расширения, но на вкладке Themes. Выбирайте по названию или щелкайте по кнопке «Подробнее», чтобы посмотреть внешний вид темы на Github.
- Чтобы тему можно было поменять, зайдите в меню Вид -> Themes...и выберите нужную. Вы сможете сразу же увидеть изменения. Останавливайтесь на той, которая наиболее приемлема для вас.





# Несколько слов о создании файлов HTML

- В системах Windows, начиная с 7-й версии обычно принято скрывать расширения файлов.



Name	Date modified	Type	Size
first.html	29.05.2017 11:30	Chrome HTML Document	2 КБ
second.html.txt	17.06.2017 22:35	Текстовый документ	0 КБ
test.html	29.05.2017 10:05	Chrome HTML Document	2 КБ

# Структура HTML-файла.

## DOCTYPE

- Любой html-файл имеет базовую структуру, которая состоит из тегов html, head и body. Но начинается она всегда с объявления типа документа — DOCTYPE.
- Мы рассматриваем синтаксис последнего на данный момент стандарта HTML — это HTML5. Для него тип документа указывается крайне просто:

```
<!DOCTYPE html>
```

- Предыдущими стандартами являются HTML4.01 и XHTML 1.0, которые существовали в различных вариантах: строгом, переходном и для фреймов, поэтому DOCTYPE необходимо было объявлять по-разному в зависимости от типа разметки, которая использовалась в html-документе.

## Строгий тип документа

- Применяется строгий синтаксис языка соответствующего стандарта, а также допускается включать все теги и атрибуты, кроме осуждаемых.

- Для HTML 4.01:

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

- Для XHTML 1.0:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//  
EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- Синтаксис языка зависят от используемой версии, но в обоих вариантах должны соблюдаться следующие правила:

- ■ Крайне нежелательно использовать такие теги, как: <applet>, <basefont>, <center>, <dir>, <font>, <isindex>, <noframes>, <plaintext>, <s>, <strike>, <u>, <xmp>, т.к. они относятся к осуждаемым (*deprecated*). Вместо них необходимо задавать форматирование с помощью стилей css.
- ■ Кроме того, нельзя добавлять в <body> любой текст, изображения и элементы форм напрямую. Все эти элементы должны находиться внутри других блочных элементов, например, <p> или <div>.
- ■ Осуждается применение таких атрибутов, как target для ссылок (тег <a>), а также start (тег <ol>), type (теги <li>, <ol>, <ul>) и др.
- ■ Еще не разрешено использовать фреймы.

## Переходный тип документа

- В этом случае используется «мягкий» синтаксис языка, также можно использовать все теги и атрибуты, включая осуждаемые. Для HTML 4.01:

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

- Для XHTML 1.0:

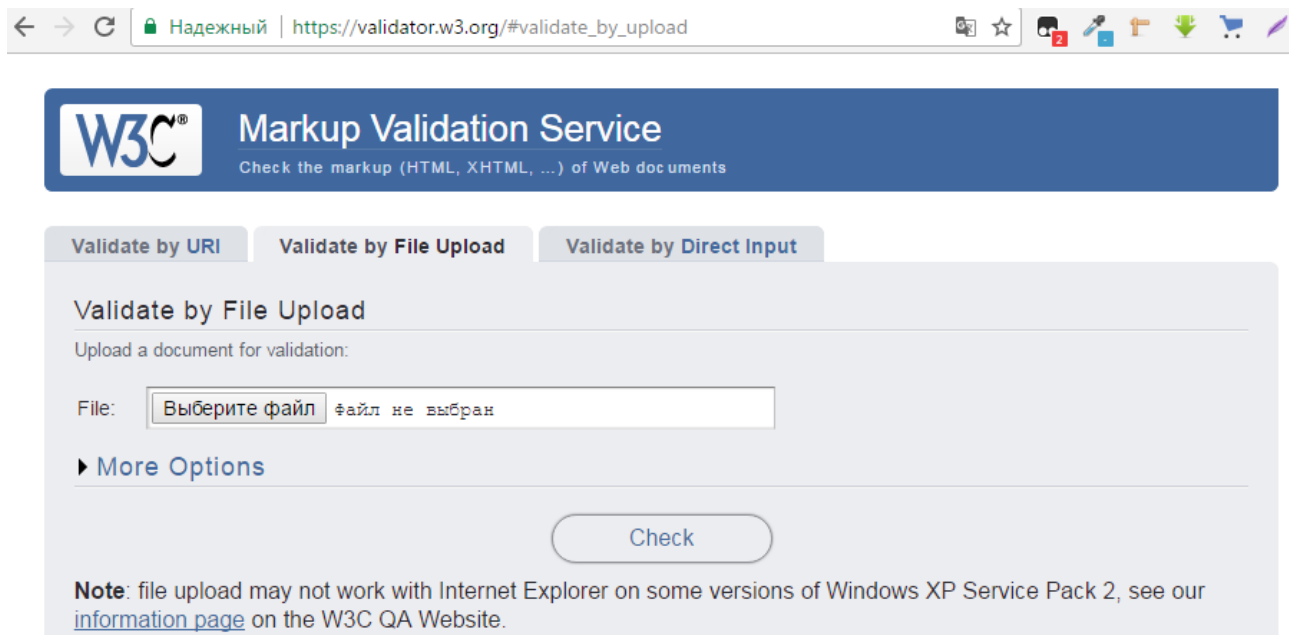
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0  
Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/  
xhtml1-transitional.dtd">
```

- Цель таких переходных доктайпов заключается в постепенном знакомстве с синтаксисом языка. В документах с переходным доктайпом можно использовать атрибут `target`, который позволяет открывать ссылку в новом окне. Также можно использовать такие теги, как `<center>` или `<font>`, которые так милы сердцу большинства начинающих html-кодеров.

- Но в этом DOCTYPE фреймы также не разрешены. Если у вас есть элементы форматирования, которые на данный момент уже устарели — такой доктайп позволит создать валидный документ. Но, забудьте о тегах `<center>` или `<font>` — такие css-свойства, как `text-align: center` или `font-family`, `font-size` и `color` дадут вам значительно больше вариантов управления внешним видом документа.

# Валидация html-документов

- Валидация html-документов — это проверка отдельных html-документов и сайтов на правильность кода.
- Доступен по ссылкам:
- ■■ <https://validator.w3.org/>
- ■■ <https://html5.validator.nu/>
- На первых порах вам больше пригодится валидатор, позволяющий загрузить файл, а не указать ссылку на сайт в Интернете. Поэтому ссылка будет такой: [https://validator.w3.org/#validate\\_by\\_upload](https://validator.w3.org/#validate_by_upload).



The screenshot shows a web browser window with the address bar displaying [https://validator.w3.org/#validate\\_by\\_upload](https://validator.w3.org/#validate_by_upload). The page header features the W3C logo and the text "Markup Validation Service" with the subtitle "Check the markup (HTML, XHTML, ...) of Web documents". Below the header, there are three tabs: "Validate by URI", "Validate by File Upload" (which is active), and "Validate by Direct Input". The "Validate by File Upload" section contains the text "Upload a document for validation:" followed by a file selection interface. It shows a "File:" label, a button that says "Выберите файл" (Select file), and the text "файл не выбран" (file not selected). Below this is a link for "More Options". At the bottom of the main content area is a large "Check" button. A note at the very bottom states: "Note: file upload may not work with Internet Explorer on some versions of Windows XP Service Pack 2, see our [information page](#) on the W3C QA Website."

# Валидация html-документов

- Вот, что выдал валидатор при загрузке документа с `<!DOCTYPE html>` и тегами `<font>` и `<center>` внутри

← → ↻ Надежный | <https://validator.w3.org/nu/#file> 📄 ☆

## Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

**Showing results for second.html**

Checker Input

Show ☐ source ☐ outline ☐ image report [Options...](#)

Check by [file upload](#) [Выберите файл](#) [Файл не выбран](#)

Uploaded files with .xhtml or .xht extensions are parsed using the XML parser.

[Check](#)

Use the Message Filtering button below to hide/show particular messages, and to see total counts of errors and warnings.

[Message Filtering](#)

- Error** The `font` element is obsolete. [Use CSS instead.](#)  
From line 13, column 1; to line 13, column 18  
`<h2><font color="red">&lt; !D`
- Error** The `center` element is obsolete. [Use CSS instead.](#)  
From line 15, column 1; to line 15, column 8  
`</font><center>Для ХН`
- Error** The `font` element is obsolete. [Use CSS instead.](#)  
From line 17, column 1; to line 17, column 20  
`</center><font color="green">&lt; !D`

# Базовая структура html-документа

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>First HTML Document</title>
</head>

<body>
  <!--      Здесь будет размещен основной код-->
</body>
</html>
```

- Корневым тегом является `<html>`. Все остальные теги вкладываются в него. У этого тега принято указывать атрибут `lang`, который отвечает за язык, на котором создана данная html-страница. В примере `lang="en"`, т.к. английский является наиболее популярным языком в мире, да и текст между тегами `<title>` тоже английский.
- Если вы создаете страницу с русским текстом, то значение этого атрибута должно быть «ru»: `<html lang="ru">`
- Далее в структуре документа идет тег `<head>`, содержимое которого не отображается на странице. Тем не менее, это важный тег, выполняющий «служебные» функции.
- Во-первых, в нем указывается `<title>` документа, который отображается на вкладке в браузере



- Во-вторых, в нем располагаются meta-теги, которые выдают информацию для поисковиков и не только. Тег `<meta charset="UTF-8">` указывает кодировку документа. Неправильное указание кодировки приведет к тому, что текст вашей страницы будет отображаться «крякозябрами».

[illegible]

- Такой вид страницы получился при указании кодировки в виде `<meta charset=»windows-1251»>` тогда как документ был сохранен в кодировке utf-8.
- **Примечание:** в стандарте HTML5 упростилось написание некоторых тегов, к которым, в том числе, относится и `<meta>`. Ранее (HTML 4.01) необходимо было указывать кодировку несколько более длинным способом:

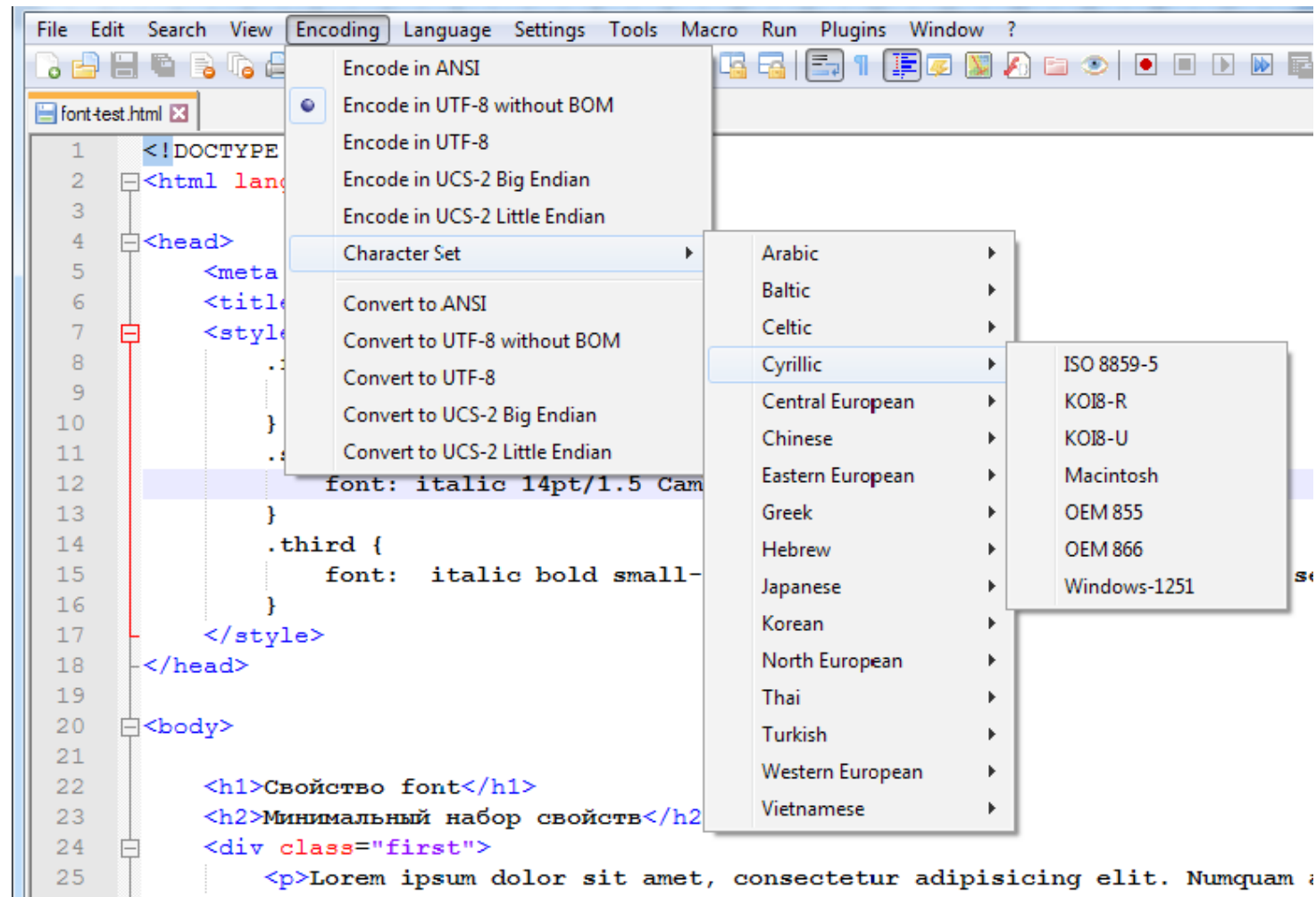
```
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
```

## Кодировка документа

- В системе Windows по умолчанию для файлов введена кодировка символов в системе ASCII, в которую включены латинские и русские символы, цифры, знаки препинания и др. В HTML ей соответствует кодировка windows-1251.
- ASCII — это однобайтная кодировка, которая позволяет получить файл меньшего размера, но и по своим возможностям уступает двухбайтной кодировке UTF-8.
- Например, вам сложно будет создать на ней многоязычный сайт, который должен отображаться на русском, английском и, например, чешском языке. В этом смысле двухбайтная кодировка UTF-8 является более универсальной, т.к. вмещает в себя коды символов на большом количестве языков. Именно поэтому она стала самой популярной в настоящее время и является сейчас стандартом «де-факто» в мире веб-разработок. Кроме указанных кодировок для русского языка есть еще ISO-8859-5 и KOI-8-R, но их мало используют.

# Кодировка документа

- Если вы установили на компьютер программу Notepad++, откройте ее и в меню Кодировки -> Кодировки -> Кириллица посмотрите доступные варианты



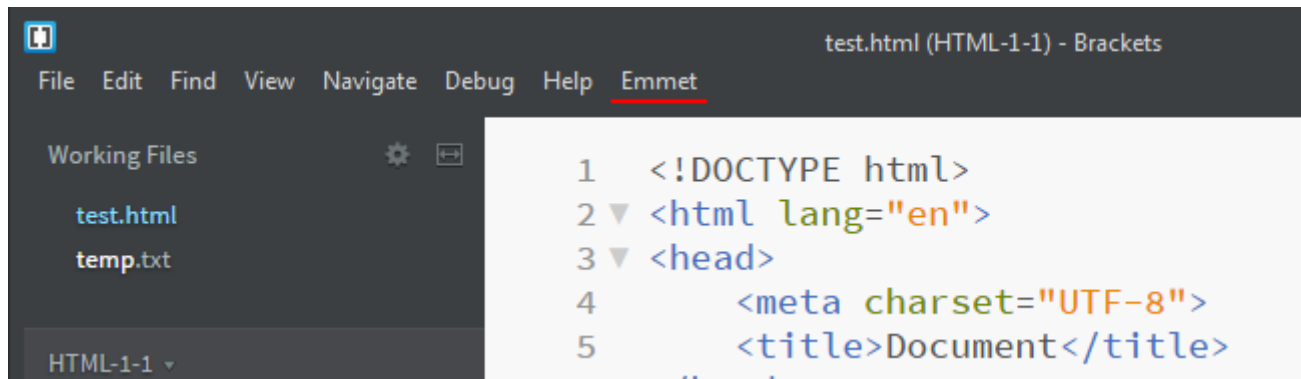
# Тело документа — тег `<body>`.

## Комментарии в HTML

- Тег **`<body>`** является тем местом, куда вы будете помещать основной html-код. Именно его содержимое является основным контентом страницы и выводится в браузер.
- Все, что будет находиться между открывающим `<body>` и закрывающим тегом `</body>`, также должно быть отформатировано в виде тегов.
- **Комментарии** в HTML не отображаются на странице.
- Весь текст, записанный между тегами
- `<!-- -->`
- будет скрыт от посетителя вашего сайта.
- В Brackets (как и в других редакторах кода, впрочем) есть быстрые клавиши для добавления комментариев — это `CTRL+ /`. Если вы хотите закомментировать одну строку, достаточно установить в любом ее месте курсор и нажать `CTRL+ /`. Комментарии автоматически добавятся в начале и в конце данной строки. Для комментирования нескольких абзацев, необходимо будет сначала их выделить, а уже потом нажать `CTRL+ /`.

# Использование плагина Emmet для создания структуры документа

- Проверить, что он установлен у вас в редакторе очень просто — посмотрите на строку меню Brackets — если в конце вы видите пункт Emmet — значит, все в порядке



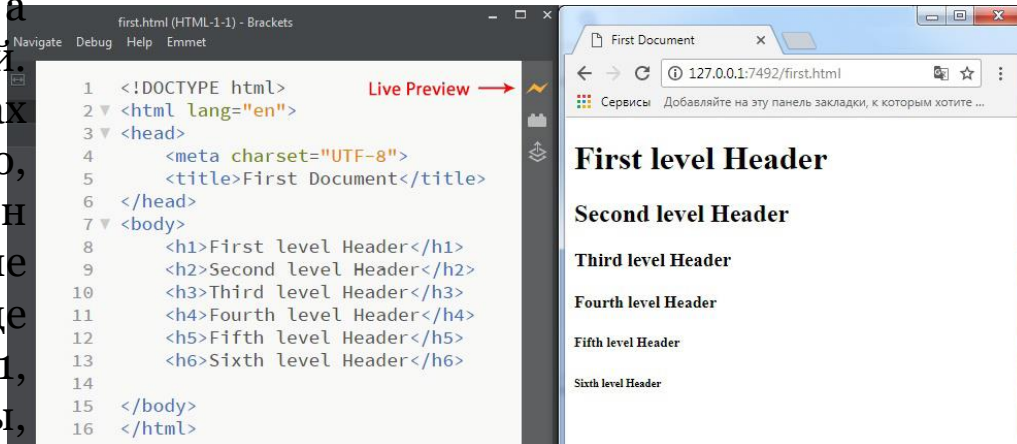
- Код базовой структуры набирается с помощью Emmet комбинацией ! и клавиша Tab.

# Теги заголовков и абзацев

- Заголовки — это обязательная часть html-документа. Есть даже 2 новых тега в HTML5, которые без заголовков не пройдут валидацию — это `<article>` и `<section>`. В HTML предусмотрено **6 уровней заголовков**, которые отличаются друг от друга размером шрифта. Обозначаются заголовки так:

```
<h1>Заголовок 1-го уровня</h1>
<h2>Заголовок 2-го уровня</h2>
<h3>Заголовок 3-го уровня</h3>
<h4>Заголовок 4-го уровня</h4>
<h5>Заголовок 5-го уровня</h5>
<h6>Заголовок 6-го уровня</h6>
```

- Буква `h` в теге обозначает, что это заголовок (от английского *heading*), а цифра — какой уровень заголовка. Чем больше цифра — тем меньше важность и размер шрифта заголовка.
- `<h1>` — самый крупный и заметный, а `<h6>` — уже почти нечитаемый. Собственно, и вес этих заголовков в глазах поисковиков тоже различен. Как правило, заголовков типа `h1` на странице должен быть один или два — это название компании и название страницы. Еще желательно, чтобы текст в `h1`, определяющим название страницы, совпадал с содержимым тега `<title>` в блоке `<head>`.



## Теги заголовков и абзацев

- Все остальные заголовки вы можете использовать по усмотрению, необязательно в порядке возрастания их уровня, но все же желательно, чтобы за `h1` шел заголовок `h2`, т.к. он тоже важен для ранжирования вашей страницы.
- **Примечание 1:** на скриншоте стрелочкой показана кнопка в виде молнии в правом верхнем углу Brackets. Это Live Preview в браузере Google Chrome, «вшитый» в редактор кода. Нажатие этой кнопки при редактировании `html`-документа позволяет в реальном времени просмотреть все изменения, происходящие в редакторе.
- Когда включен Live Preview, в браузере синей рамочкой выделен элемент, на котором сейчас расположен курсор (на скриншоте это `h2`). Если вы будете перемещать курсор выше или ниже, например, клавишами `↑` или `↓`, то увидите, как перемещается рамка с одного элемента на другой.
- **Примечание 2:** если в Brackets вам приходится создавать блоки с похожим или идентичным содержанием, имеет смысл воспользоваться сочетанием клавиш `CTRL + D` (от англ. *duplicate* — дублировать). Если курсор находится в любом месте строки, то будет продублирована вся строка. Если вы выделите часть слов или несколько строк, то продублируется весь выделенный контент.



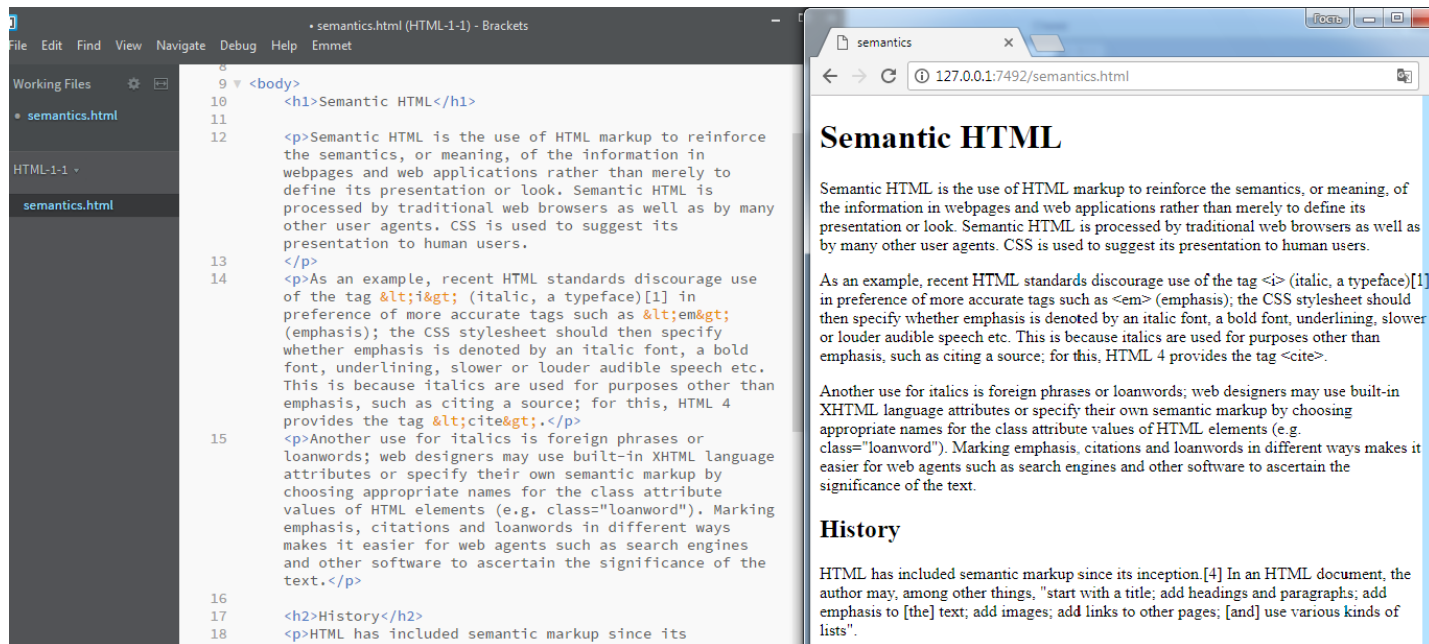
## Что такое блочные элементы?

- Блочными называются такие элементы HTML, которые по умолчанию занимают все доступное пространство внутри браузера или родительского элемента, даже если их содержимое совсем небольшое. Т.е. даже при 2-3 словах внутри заголовка он будет занимать все пространство до правой границы браузера, и текст, размещенный рядом с закрывающим тегом, будет перенесен на следующую строку.
- На скриншоте в примере видно, что заголовок h1, выделенный в Brackets в режиме Live Preview, обведен голубой рамкой и занимает все место слева направо в браузере. А текст за ним перенесен на следующую строку



# Абзацы в HTML

- Абзацы являются основными элементами для форматирования текста, который составляет статью. Для создания абзацев предназначен тег `<p></p>` (от англ. *paragraph*).
- На скриншоте видно, что текст, помещенный в абзацы, выводится в браузере обычным начертанием (а не жирным, как у заголовков), но при этом каждый абзац имеет отступы сверху и снизу. Это визуально отделяет один смысловой блок от другого



## Несколько слов об атрибутах

- Для любого тега можно задать атрибуты — дополнительные параметры в виде пар атрибут= "значение", которые в некоторой степени выделяют его среди других таких же. Атрибуты записываются только в открывающем теге и отделяются от названия тега и друг от друга пробелами. В общем случае :

```
<элемент атрибут1='значение' атрибут2='значение'  
атрибут3='значение'>Текст элемента</элемент>
```

- Пример*

```
<h2 title='Статья о пользе витаминов для детей'  
id='article1' class='article-header'> Витамины для детей </h2>
```

- В этом примере для заголовка 2-го уровня использованы универсальные атрибуты title, id и class, т.е. такие, которые можно добавить для любого элемента. Еще к универсальным относятся tabindex, data-атрибуты и некоторые другие, редко используемые, например, contenteditable, hidden или contextmenu. Но также существуют атрибуты, которые характерны только для определенных тегов. Например, для встраивания изображения нужен тег с атрибутами src и alt, которые встречаются характерны только для него (хотя src нужен еще для таких тегов, как <script> и <iframe>):

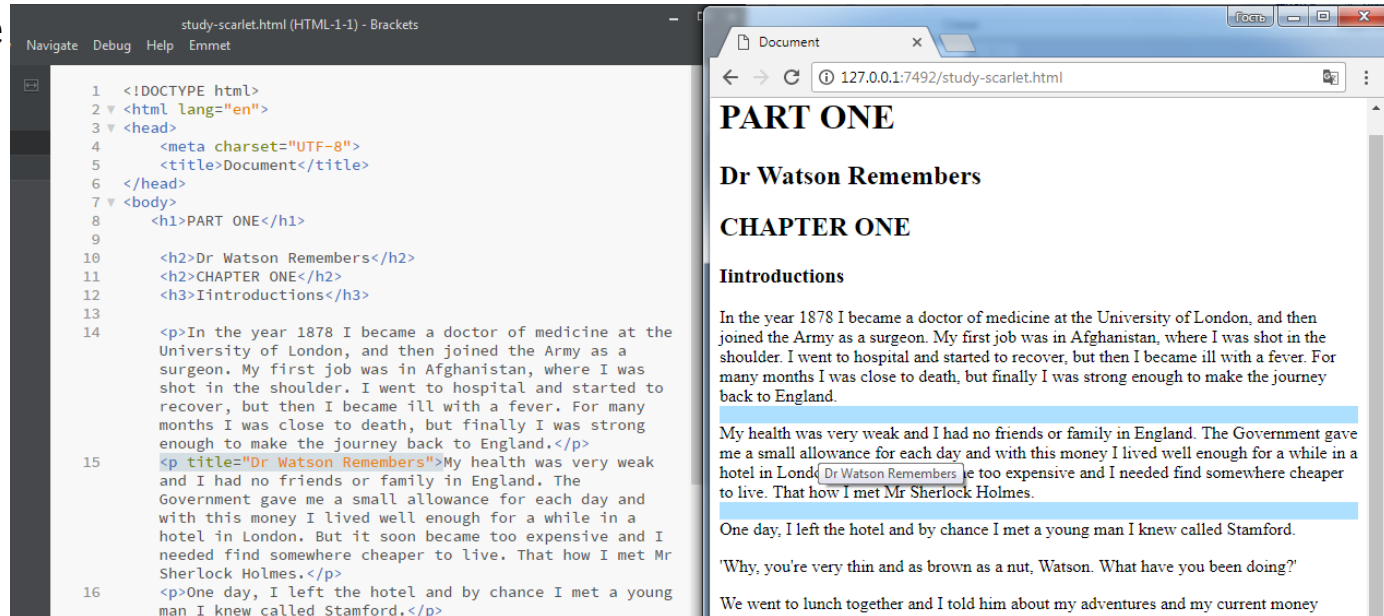
```
<img src='images/photo.jpg' alt='Картинка'>
```

# Абзацы в HTML

- При добавлении атрибутов нужно соблюдать несколько простых правил:
- 1. Атрибуты отделяются друг от друга пробелами.
- 2. Значения атрибутов записываются в двойных или одинарных кавычках.
- 3. Если внутри значения атрибута необходимо использовать кавычки или апостроф, необходимо сочетать оба вида кавычек с учетом их вложенности:

```
<p title='Абзац из книги Б. Акунина "Азazelь"'>
```

- Здесь значение атрибута title взято в одинарные кавычки, а внутри использованы двойные для названия книги. Этот атрибут добавляет всплывающую подсказку к элементу, которая появляется при наведении на элемент в браузере



## Абзацы в HTML

- 4. Если значение атрибута можно представить в виде логической величины `true` (истина) или `false` (ложь), или по другому «есть» или «нет», то записывать его можно в нескольких вариантах, которые равнозначны:

```
<input type="checkbox" checked>  
<input type="checkbox" checked="">  
<input type="checkbox" checked="checked">
```

- Примечание:** Все браузеры настроены на максимально правильное отображение тегов и их атрибутов, поэтому включают атрибут даже при наличии в нем значений вроде `true` или `1`, которые на самом деле недопустимы. Лучше все-таки избегать таких вариантов, т.к. они противоречат спецификации HTML5.

- Устаревшие атрибуты. Например, атрибут `align` позволяет выравнивать текст в абзаце, заголовке или в другом блочном элементе по левому или правому краю, по ширине элемента или по центру (`align="left"`, или `align="right"`, или `align="justify"`, или `align="center"`). Со стандартом HTML5 этот атрибут несовместим. И валидатор выдаст следующее сообщение: «Атрибут `align` для элемента `p` устарел. Вместо него используйте CSS»

1. **Error** The `align` attribute on the `p` element is obsolete. [Use CSS instead.](#)

From line 28, column 5; to line 28, column 21

```
<./p>↵ <p align="right">Лицо м
```

## Что такое Lorem Ipsum?

- Далеко не всегда в вашей практике придется форматировать готовый текст, предоставленный заказчиком. Кроме того, вы можете создавать шаблоны сайтов, о реальном содержании которых на этапе разработки можете только догадываться. Поэтому вам наверняка пригодится шаблонный текст, который начинается словами *Lorem ipsum*. Его, кстати, еще называют «текст-рыба». Т.е. это текст, который помогает заполнить сайт контентом, но при этом смысла не имеет.
- **Цитата из Википедии:** *Lorem ipsum* — классическая панграмма, условный, зачастую бессмысленный текст-заполнитель, вставляемый в макет страницы. Используется в качестве заполнителя по крайней мере с XVI века[1]. Является искажённым отрывком из философского трактата Марка Туллия Цицерона «О пределах добра и зла», написанного в 45 году до н. э. на латинском языке.
- При соответствующем запросе Google предоставляет доступ к массе генераторов Lorem ipsum:
  - <http://generator.lorem-ipsum.info/>;
  - <http://lorem-ipsum.perbang.dk/>;
  - <http://www.blindtextgenerator.com/ru>.

## Использование Emmet для генерации Lorem Ipsum

- Чтобы получить абзац с текстом-рыбой, сразу после открывающего тега `<body>` или после уже существующего в нем другого тега, наберите аббревиатуру `p>lorem`

и нажмите клавишу Tab.

- По умолчанию, Emmet добавит 30 слов текста. Но вы можете разнообразить количество слов, добавляя нужную цифру после `lorem`:

```
p>lorem10
```

- Получим

```
<p>
  Lorem ipsum dolor sit amet, consectetur
  adipisicing elit. Necessitatibus, possimus!
</p>
```

- Примечание:** плагин *Emmet* использует аббревиатуры — последовательности символов, которые позволяют вывести теги, атрибуты и текст в определенной последовательности и с определенными уровнями вложенности.
- Важно понимать, что внутри аббревиатуры не должно быть пробелов (исключение — текст внутри фигурных скобок). А в самом конце обязательно нужно нажать клавишу Tab, чтобы раскрыть аббревиатуру и получить отформатированный по правилам html текст. Опять-таки после текста аббревиатуры не должно быть пробела, и курсор должен находиться именно в конце, а не где-нибудь в середине набранного текста.

# Использование Emmet для генерации Lorem Ipsum

```
h1{First heading}+h2{Second heading}+p*3>lorem20
```

```
<h1> Firstheading </h1>
<h2> Second heading </h2>
<p>
Lorem ipsum dolor sit amet, consectetur dipisicing elit. Placeat
corporis quasi
perspiciatis? Aperiam eveniet dolorem culpa
distinctio, rem quos adipisci.
</p>
<p>
Debitis, dolorem! Sunt autem veritatis magnam!
Ipsa, dolorem harum laborum praesentium,
quas unde ab, alias saepe ullam similique nulla
beatae.
</p>
<p>
Quis soluta saepe incidunt voluptas consequuntur
iste repellat, quasi quos provident, nostrum, a.
Repellendus aspernatur, veritatis ea cum aliquid
architecto!
</p>
```



# Что такое Lorem Ipsum?

```
h1>lorem4^h2>lorem5^p*2>lorem15^h3>lorem4^p*3>lorem10
```



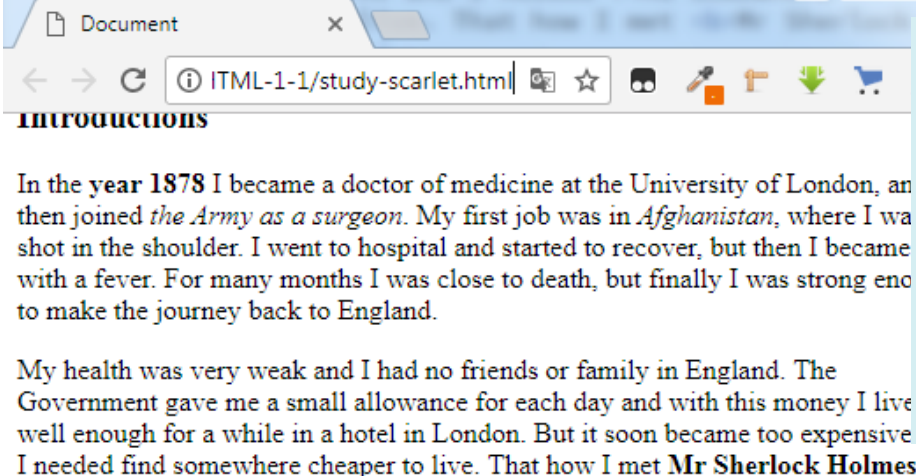
# Вложенные теги

```

study-scarlet.html (HTML-1-1) - Brackets
gate Debug Help Emmet
12 <h3>Dr Watson Remembers</h3>
13 <h2>CHAPTER ONE</h2>
14 <h3>Introductions</h3>
15
16 <p>In the <b>year 1878</b> I became a doctor
of medicine at the University of London, and
then joined <i>the Army as a surgeon</i>. My
first job was in <i>Afghanistan</i>, where I
was shot in the shoulder. I went to hospital
and started to recover, but then I became ill
with a fever. For many months I was close to
death, but finally I was strong enough to make
the journey back to England.</p>
17 <p>My health was very weak and I had no
friends or family in England. The Government
gave me a small allowance for each day and
with this money I lived well enough for a
while in a hotel in London. But it soon became

```

- Чтобы выделить часть текста жирным или курсивным шрифтом, можно использовать теги `<b>` или `<i>`



```

<p> In the <b>year 1878</b> I became a doctor of
medicine at the University of London, and then
joined <i>the Army as a surgeon</i>. My first job
was in <i>Afghanistan</i>, where I was shot in
the shoulder.
I went to hospital and started to recover, but
then I became ill with a fever. For many months
I was close to death, but finally I was strong
enough to make the journey back to England.
</p>

```

## Вложенные теги

- Тег `<b>` **bold** (*жирный*), а `<i>` — **italics** (*курсив*), использование этих тегов приводит к выделению текста между ними с помощью жирного или курсивного начертания. Эти теги физического форматирования. В отличие от блочных тегов заголовков и абзацев эти теги являются строчными.
- Поэтому оборачивают в такие теги не все предложение или блок текста, а какую-то его часть, например, 2-3-4 слова, как в примере на скриншоте.
- Таким же образом работают еще 2 тега — `<strong>` и `<em>`, т.е. `<strong>` выделяет текст жирным, а `<em>` — курсивом, но эти 2 элемента относятся к группе тегов логического форматирования. В контексте разметки страницы они выполняют роль «усилителя внимания», т.е. призваны показать (в большей степени поисковикам, чем посетителям), что выделенный в них текст важен для пользователя. Но и теги физического, и теги логического форматирования находятся внутри блочных элементов (в примере это абзац — `<p>`), т.е. являются вложенными или дочерними по отношению к абзацу. А абзац по отношению ко вложенным тегам является родительским элементом.
- Для вложенных тегов важно соблюдать «правило матрешки»: тег, который открыт самым первым, должен быть закрыт самым последним:

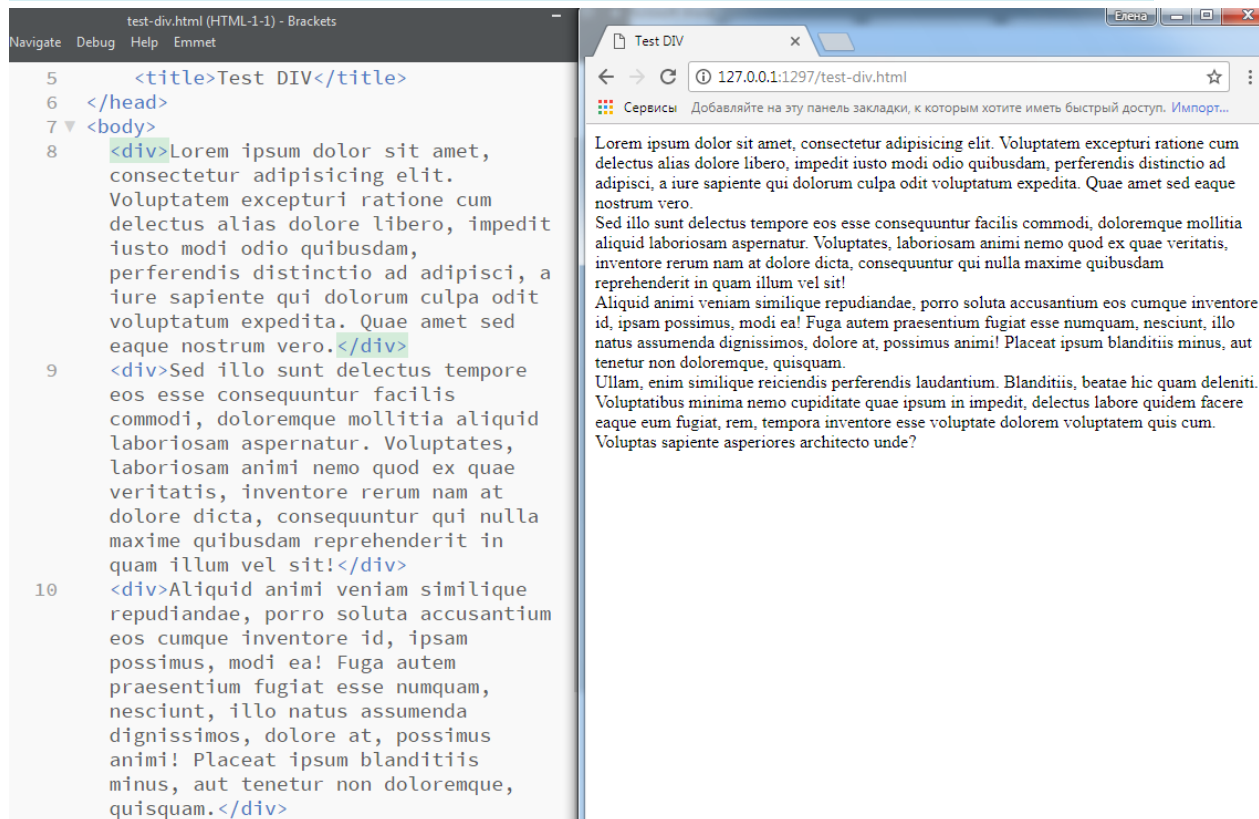
```
medicine at the University of London, and then
joined <i>the Army as a surgeon</i>. My first
job was in <b><i>Afghanistan</i></b>, where I
was shot in the shoulder. I went to hospital and
started to recover, but then I became ill with a
```

# Теги div и span

- Из заголовков и абзацев не построишь полноценную страницу. Одними из самых используемых элементов являются **div**-ы (от англ. *division* — *раздел*). Это блочные элементы, которые, в отличие от абзацев не имеют отступа сверху и снизу. Это элементы, из которых строится внутренняя структура сайта.

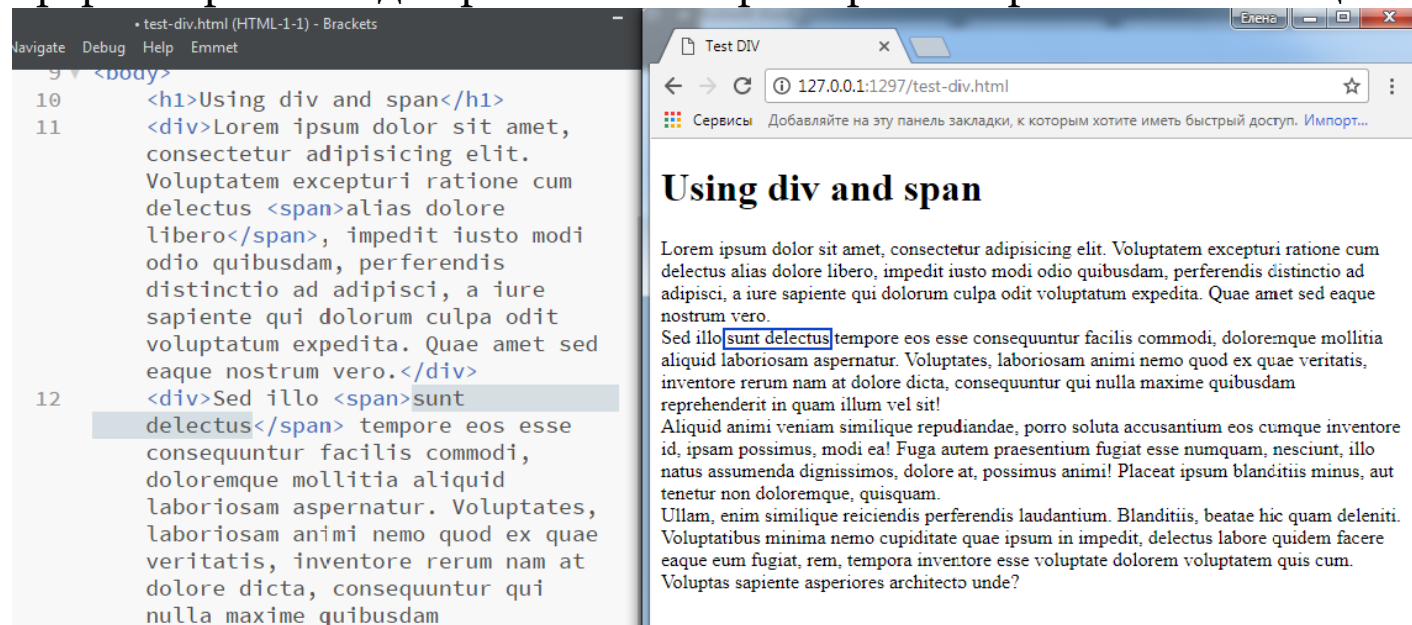
```
<div>текст</div>
```

```
div*4>lorem40
```



## Теги div и span

- Визуально тег `<div>` не слишком-то замечательно выглядит — масса текста без особого форматирования. Но этим он и хорош — для него можно задать то **CSS**-форматирование, которое необходимо для данной конкретной ситуации.
- И обычно для `div`-ов задают атрибут `class`, который и позволяет разнообразить внешний вид этих элементов. Но об этом чуть позже, когда будем рассматривать стили `CSS`.
- Также очень часто внутри текста используют теги `<span></span>` (от англ. *интервал*). Это строчные теги, которые предназначены для объединения небольшого количества текста с целью задания для него общего форматирования. Практически обязательным в этом случае для тега `span` является атрибут `class`. Именно он позволяет сделать форматирование для различных `<span>` разнообразным с помощью правил `CSS`



Sed pariatur hic eveniet quas dolores itaque maxime voluptatibus *deserunt velit quae* id, obcaecati qui sunt. Tempora voluptatibus consequatur odit, debitis cumque laborum iste dolorum eligendi eveniet nihil totam vel. *Natus assumenda* dolor excepturi quo nihil recusandae, eum debitis obcaecati?



# Тег blockquote

- Еще из блочных элементов рассмотрим тег `blockquote`, который предназначен для форматирования чьих-либо высказываний — от известных людей до директоров компаний, для которых создается сайт.

```
<blockquote>Блочная цитата</blockquote>
```

- По умолчанию, `blockquote` имеет отступы в 40px справа и слева, а также отступы сверху и снизу, как в абзацах

The image shows a side-by-side comparison of HTML code and its rendered output. On the left, a code editor (Brackets) displays the HTML for a file named 'use-blockquote.html'. The code includes a head section with a UTF-8 charset and a title 'Use blockquote', followed by a body section with a heading 'Using blockquote tag' and several paragraphs, including a blockquote from Thomas Edison and a quote from Jim Morrison. On the right, a web browser window shows the rendered page. The blockquote is correctly formatted with indentation on both sides and space before and after the paragraph. The text inside the blockquote is indented further, and the quote from Jim Morrison is separated by a line break.

```

use-blockquote.html (HTML-1-1) - Brackets
Debug Help Emmet
4 <head>
5   <meta charset="UTF-8">
6   <title>Use blockquote</title>
7 </head>
8
9 <body>
10  <h1>Using blockquote tag</h1>
11  <p>The <b>blockquote tag</b> specifies
    a section that is quoted from another
    source.</p>
12  <p>Browsers usually indent blockquote
    elements.</p>
13  <blockquote>
14    For 50 years, WWF has been
    protecting the future of nature.
    The world's leading conservation
    organization, WWF works in 100
    countries and is supported by 1.2
    million members in the United
    States and close to 5 million
    globally.
15  </blockquote>
16  <p>The tag <b>blockquote</b> is for
    example used to display the statements
    of famous people.</p>
17  <p>Thomas Edison says:</p>
18  <blockquote>Success is one percent
    inspiration, ninety-nine percent
    perspiration.
19    <br>Thomas Edison/
20  </blockquote>

```

Use blockquote

127.0.0.1:2255/use-blockquote.html

Сервисы Добавляйте на эту панель закладки, к которым хотите иметь быстрый доступ. Им...

## Using blockquote tag

The **blockquote** tag specifies a section that is quoted from another source.

Browsers usually indent blockquote elements.

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

The tag **blockquote** is for example used to display the statements of famous people.

Thomas Edison says:

Success is one percent inspiration, ninety-nine percent perspiration.  
/Thomas Edison/

Jim Morrison about freedom:

The most important kind of freedom is to be what you really are. You trade in your reality for a role. You trade in your sense for an act. You give up your ability to feel, and in exchange, put on a mask. There can't be any large-scale revolution until there's a personal revolution, on an individual level. It's got to happen inside first.  
/Jim Morrison/

6, Column 10 — 29 Lines    UTF-8    HTML    Spaces: 4

## Одиночные теги

- В примере с `blockquote` внутри цитаты был использован тег `<br>` — от англ. *break* — *разбивать*. Он предназначен для переноса текста, следующего за ним, на следующую строку и является строчным, т.е. обычно размещается в тексте абзацев, `div`-ов и других элементов.
- Поскольку тег не имеет внутреннего содержимого, то закрывающий тег ему не нужен, т.е. он относится к группе одиночных тегов, или тегов без содержимого.
- К ним же относится тег `<img>` для встраивания изображений и целая группа тегов `<input>`, из которых состоят формы.
- Здесь же мы рассмотрим еще один такой тег — это `<hr>`, или горизонтальная строка (от англ. *horizontal row*).
- Собственно, с его помощью мы можем вывести в браузер горизонтальную линию. В HTML4.01 для нее можно было задать ряд атрибутов, которые в HTML5 являются отмененными. Поэтому все «красивости» нужно задавать через `css`.
- **Примечание:** В XHTML для всех одиночных тегов был обязательным закрывающий слэш в конце через пробел от имени тега, поэтому на ряде сайтов вы можете встретить такое написание тегов



# Одиночные теги

use-blockquote.html (HTML-1-1) - Brackets

Navigate Debug Help Emmet

```

10 <p>The <b>blockquote tag</b>
    specifies a section that is quoted
    from another source.</p>
11 <p>Browsers usually indent
    blockquote elements.</p>
12 <hr>
13 <blockquote>
14     For 50 years, WWF has been
    protecting the future of
    nature. The world's leading
    conservation organization, WWF
    works in 100 countries and is
    supported by 1.2 million
    members in the United States
    and close to 5 million
    globally.
15 </blockquote>
16 <p>The tag <b>blockquote</b> is for
    example used to display the
    statements of famous people.</p>
17 <p>Thomas Edison says:</p>
18 <hr>
19 <blockquote>Success is one percent
    inspiration, ninety-nine percent
    perspiration.
20     <br>/Thomas Edison/
21 </blockquote>
22
23 <p>Jim Morrison about freedom:</p>
24 <hr>
25 <blockquote>
26     The most important kind of
    freedom is to be what you
    really are. You trade in your
  
```

Line 16, Column 19 — 31 Lines

INS UTF-8 HTML Spaces: 4

Use blockquote

127.0.0.1:1297/use-blockquote.html

Сервисы Добавляйте на эту панель закладки, к которым хотите иметь быстрый доступ. Импорт...

The **blockquote** tag specifies a section that is quoted from another source.

Browsers usually indent blockquote elements.

For 50 years, WWF has been protecting the future of nature. The world's leading conservation organization, WWF works in 100 countries and is supported by 1.2 million members in the United States and close to 5 million globally.

The tag **blockquote** is for example used to display the statements of famous people.

Thomas Edison says:

Success is one percent inspiration, ninety-nine percent perspiration.  
/Thomas Edison/

Jim Morrison about freedom:

The most important kind of freedom is to be what you really are. You trade in your reality for a role. You trade in your sense for an act. You give up your ability to feel, and in exchange, put on a mask. There can't be any large-scale revolution until there's a personal revolution, on an

**Спасибо за внимание.**