

# Учебник HTML

# Введение

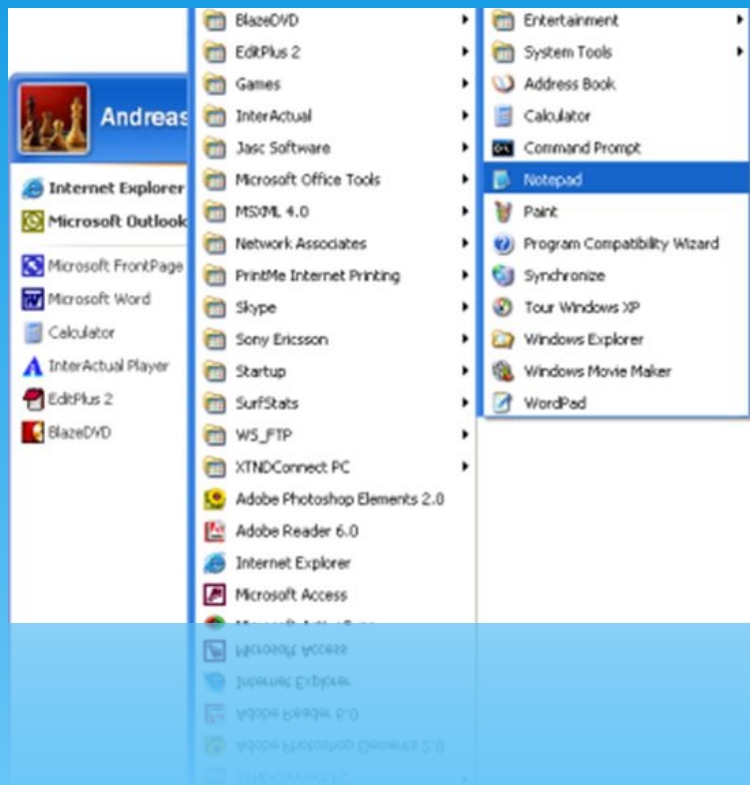
- Многие думают, что создать web-сайт очень трудно. Это совершенно не так! Каждый может научиться создавать web-сайты. Если вы готовы, то сможете сделать сайт уже через час.
- Другое заблуждение - также ошибочное, что для создания сайтов необходимо иметь дорогое и сложное программное обеспечение. Действительно, есть множество программ, которые рекламируются как необходимые для создания сайтов. Но, если вы хотите делать всё правильно, вы должны всё сделать сами. К счастью это весьма просто, и у вас уже есть все необходимые программы.
- Цель данного учебника - дать вам простое и корректное понимание того, как делать web-сайты. Это учебник начинается с нуля и не требует абсолютно никаких предварительных навыков программирования.
- Мы не можем рассказать здесь обо всём. От вас потребуются немного настойчивости и желания поэкспериментировать. Но не сомневайтесь - создание web-сайтов вам понравится и принесёт громадное удовлетворение результатами вашей работы.
- То, как вы будете использовать этот учебник, зависит от вас. Но мы советуем читать не более двух-трёх уроков в день и выделять время для экспериментов с тем новым, что вы узнаете в каждом уроке.
- Okay, начнём!

В этом первом уроке мы кратко представим вам утилиты, которые нужны для создания web-сайта.

# Урок 1: Начнём

# Что же необходимо иметь?

- \* Вероятнее всего у вас уже всё необходимое.
- \* У вас есть "браузер/browser". Это программа просмотра web-страниц. Сейчас вы просматриваете данную страницу в вашем браузере.
- \* Не важно, какой браузер у вас имеется. Обычно это Microsoft Internet Explorer. Но есть и другие - Opera и Mozilla Firefox, и их также можно использовать.
- \* Возможно вы уже слышали, или даже использовали такие программы, как Microsoft FrontPage, Macromedia Dreamweaver или даже Microsoft Word, которые могут - или заявляют что могут - создавать web-сайты для вас. Пока что забудьте об этих программах! Они не научат вас тому, как кодировать web-сайт.



Вам нужен простой текстовый редактор. Если вы используете Windows, то в ней есть Notepad, который обычно находится в меню Старт, Программы, Стандартные

- \* Если вы используете не Windows, то можете выбрать аналогичный простой текстовый редактор. Например, Pico (Linux) или TextEdit (Mac).
- \* Notepad прекрасно подходит для создания кода, поскольку никак не влияет на то, что вы пишете. Это даёт вам полный контроль над текстом. Проблема многих навороченных программ в том, что в них много стандартных функций, из которых вы можете выбирать необходимые. Обратной стороной этого является то, что всё подгоняется под эти стандартные функции. Поэтому такие программы часто не в состоянии создать точно такой сайт, какой необходим вам. Или что ещё хуже, они начинают изменять написанный вами код. При использовании Notepad или иного простого текстового редактора только вы отвечаете за то, что написано.
- \* Браузер и Notepad (или аналогичный простой редактор) - вот всё, что вам необходимо для работы с данным учебником и создания своих web-сайтов.

# Нужно ли мне быть в сети online?

- \* Вам не обязательно иметь соединение с Internet - ни для чтения учебника, ни для создания ваших web-сайтов.
- \* Вы можете даже распечатать этот учебник, или просто отключиться от Internet. Вы можете делать web-сайт на жёстком диске вашего компьютера и выгружать его в Internet после окончания работы.

# Что дальше?

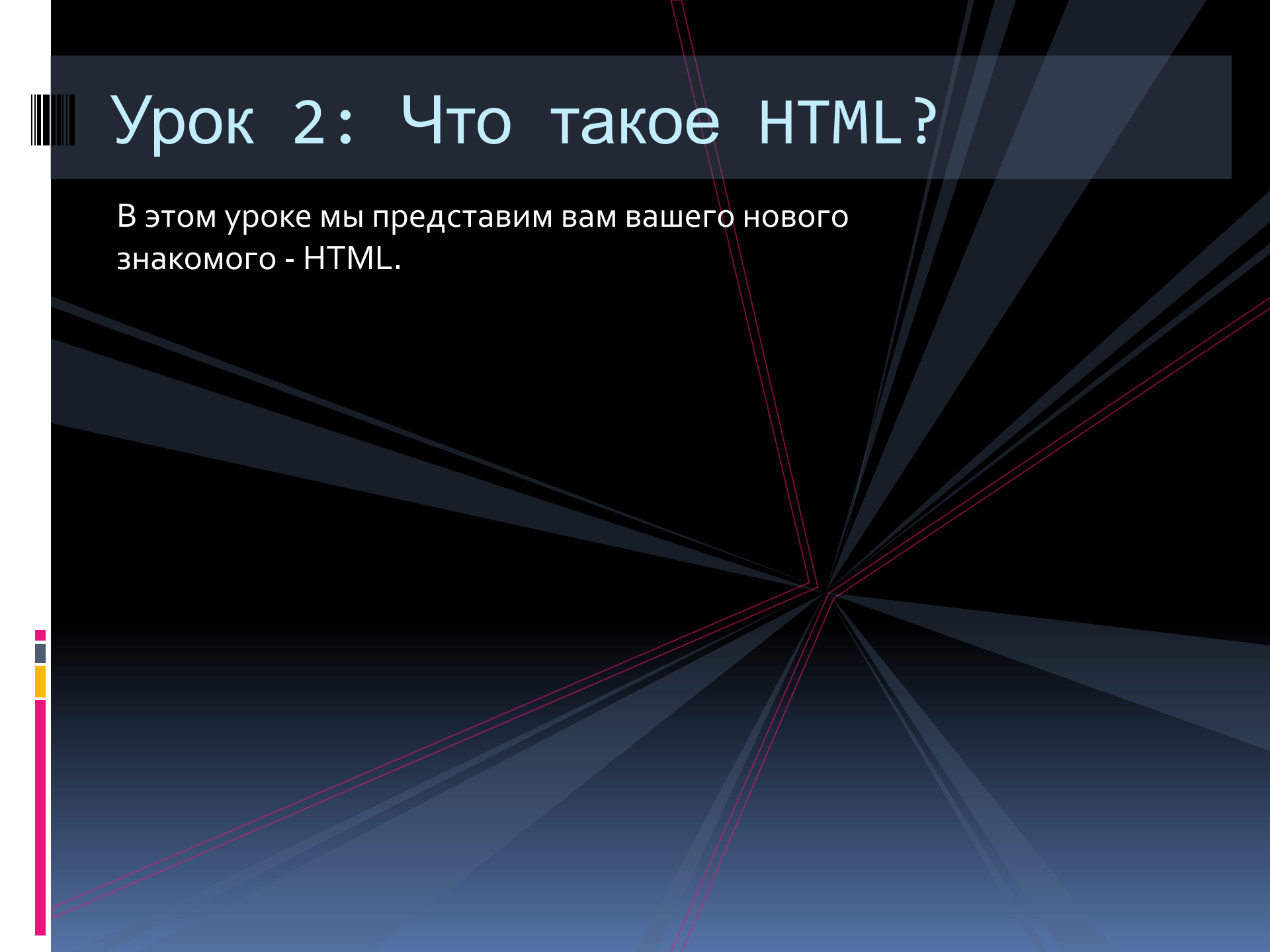
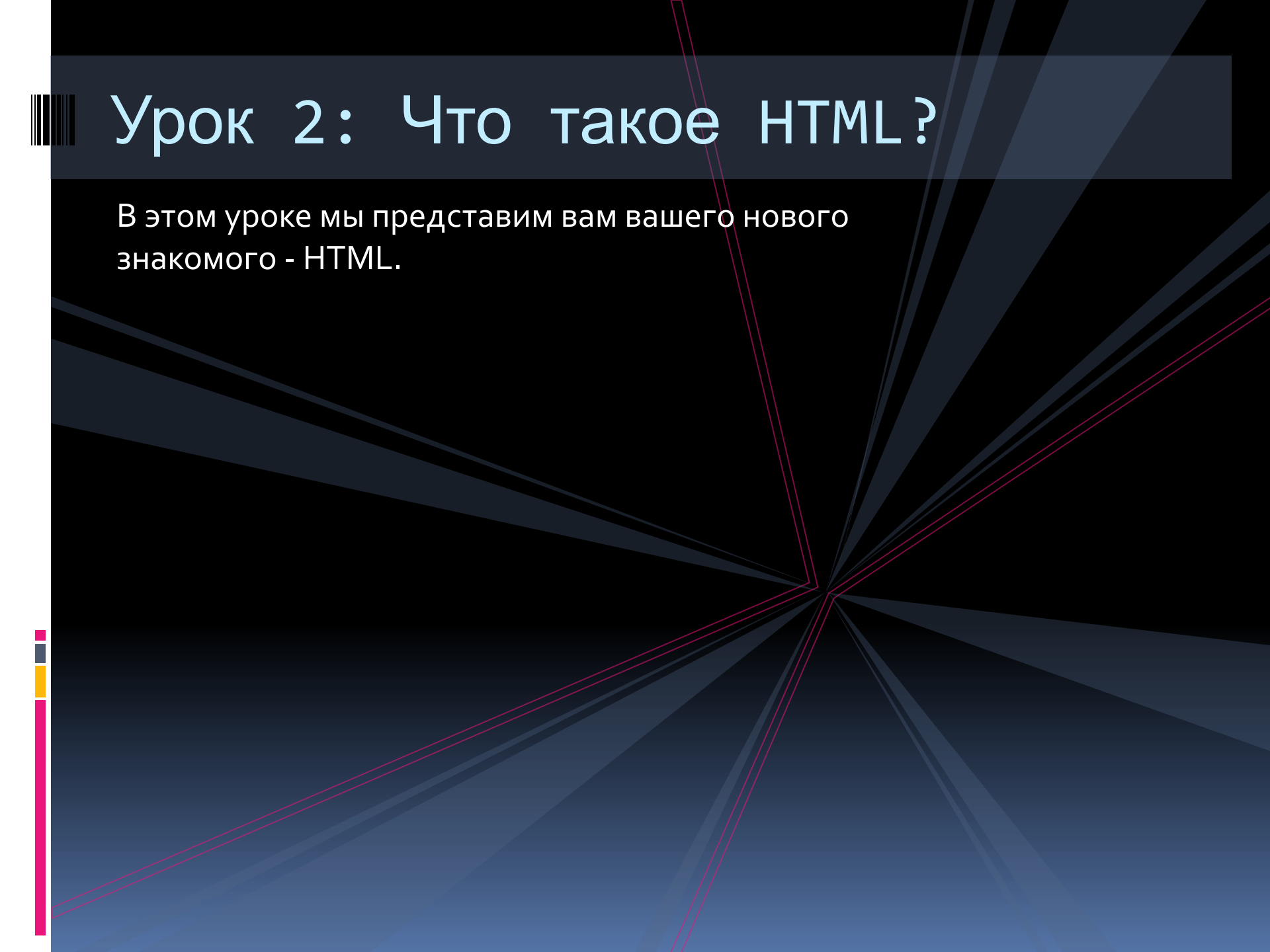
- \* Переходите к следующему уроку и прочтите о HTML, прежде чем реально начать его изучение в Уроке 3.





# Урок 2: Что такое HTML?

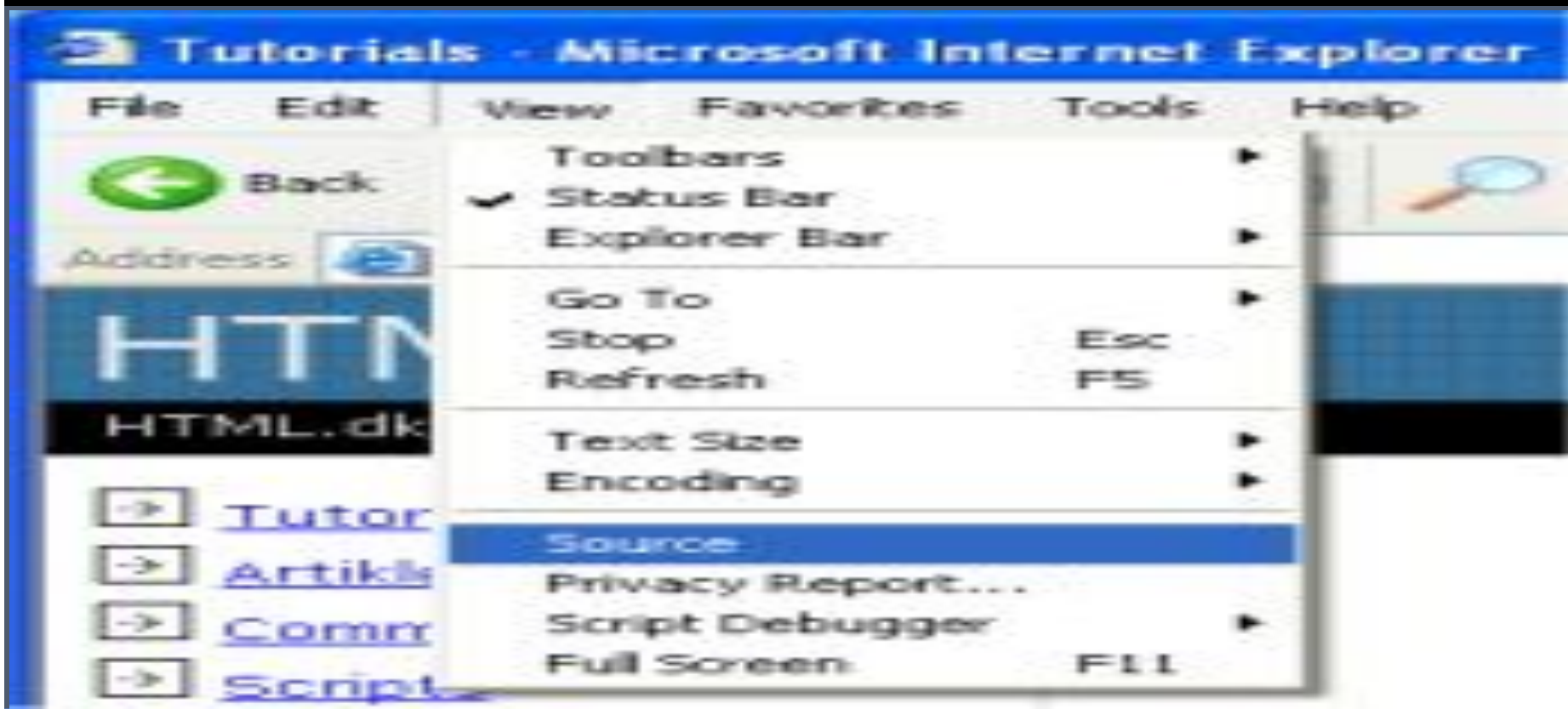
В этом уроке мы представим вам вашего нового знакомого - HTML.




# Что такое HTML?

- HTML это "родной язык" вашего браузера (программы просмотра вэб-страниц).
- Говоря кратко, HTML был изобретён в 1990 году учёным, Тимом Бёрнсом-Ли (Tim Berners-Lee), и предназначался для облегчения обмена документами между учёными различных университетов. Проект имел больший успех, чем Tim Berners-Lee мог ожидать. Этим изобретением HTML он заложил основы современной сети Internet.

HTML это язык, который позволяет представлять информацию (например, научные исследования) в Internet. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста. Чтобы увидеть HTML-коды страницы в Internet, щёлкните "View" в линейке меню вашего браузера и выберите "Source".





для неискушённого HTML-код  
выглядит сложным, но этот  
учебник поможет вам  
разобраться в нём


# Для чего я могу использовать HTML?

- Если вы хотите создавать web-сайты, вы не обойдётесь без HTML. Даже если вы используете для создания web-сайтов такие программы, как Dreamweaver, знание основ HTML значительно упростит вам жизнь, а ваш web-сайт станет намного интересней. Хорошей новостью является то, что HTML легко изучать и использовать. Через два урока вы уже создадите ваш первый web-сайт.
- HTML используется для создания web-сайтов. И это очень просто!

# Окау, но как

## расшифровывается Н-Т-М-Л?

- HTML это сокращение от "HyperText Mark-up Language/язык гипертекстовой разметки" - и на данном этапе этого знания вам вполне достаточно. Однако, чтобы всё было путём, давайте разъясним некоторые детали.
- Hyper противоположно linear/построчно. В добрые старые времена - когда кошки ловили мышей - компьютерные программы работали построчно: программа выполняла одну строку, затем переходила к выполнению следующей, и т. д. Но HTML работает по-иному - вы можете перейти куда и когда захотите. Например, не нужно посещать MSN.com до того, как посетить HTML.net.
- Text - всё понятно.
- Mark-up - это разметка, то, что вы делаете с текстом. Вы размечаете текст так же, как вы делаете это в текстовых редакторах: выставляете заголовки, списки, выделяете текст жирным шрифтом и т. д.
- Language это язык - HTML. В нём используется много английских слов.
- В данном учебнике вы будете изучать так называемы XHTML (Extensible HyperText Mark-up Language), который, коротко говоря, является новым и более структурированным вариантом HTML.



Если вы поняли, что такое HTML (и XHTML), давайте перейдём к тому, для чего они оба предназначены: к созданию web-сайтов.

# \*Урок 3: Что такое HTML-тэги?

Теперь вы готовы начать изучение HTML-тэгов.



- \* Тэги это метки, которые вы используете для указания браузеру, как он должен показывать ваш web-сайт.
- \* Все тэги имеют одинаковый формат: они начинаются знаком "<" и заканчиваются знаком sign ">".
- \* Обычно имеются два тэга - открывающий: <html> и закрывающий: </html>. Различие в том, что в закрывающем имеется слэш "/".
- \* Всё содержимое, помещённое между открывающим и закрывающим тэгами, является содержимым тэга.
- \* И, как говорится, из каждого правила есть исключения, и в HTML также имеются тэги, которые являются и открывающими, и закрывающими. Эти тэги не содержат текста, а являются метками, например, перенос строки выглядит так: <br />.
- \* HTML - это тэги, и ничего кроме тэгов. Для изучения HTML нужно изучить различные тэги.

\* "Тэги"?

\*Окау, тэг <b> информирует браузер, что весь текст между <b> и </b> должен быть напечатан жирным шрифтом. ("b" это сокращение для "bold".)

**\* Можете вы привести  
какие-нибудь  
примеры?**

## \* Пример 1

\* `<b>This text must be bold.</b>`

\* будет выглядеть при просмотре в браузере:

\* **This text must be bold.**

\* Тэги `<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>` и `<h6>` указывают браузеру создавать заголовки (h для "heading"), где `<h1>` это заголовок первого уровня - самый крупный шрифт, `<h2>` - заголовок второго уровня - шрифт меньшего размера, и `<h6>` - заголовок шестого уровня - самого низкого в этой иерархии, и самый маленький шрифт.

\* **Пример 2:**

\* `<h1>Это заголовок</h1> <h2>Это подзаголовок</h2>`

\* будет выглядеть в браузере:

\* **Это заголовок**

\* Это подзаголовок

\* Большинству браузеров безразлично, в каком регистре введены буквы тэгов. <HTML>, <html> или <HtMl> обычно даёт одинаковый результат. Однако корректным будет нижний регистр. Поэтому привыкайте печатать тэги в нижнем регистре.

**\* В каком регистре  
должны вводиться  
буквы тэгов?**

\* Вы печатаете ваши тэги в HTML-документе. На web-сайте имеется один или более HTML-документов. Когда вы бродите по Web, вы открываете различные HTML-документы.

**\* Где размещать все  
эти тэги?**

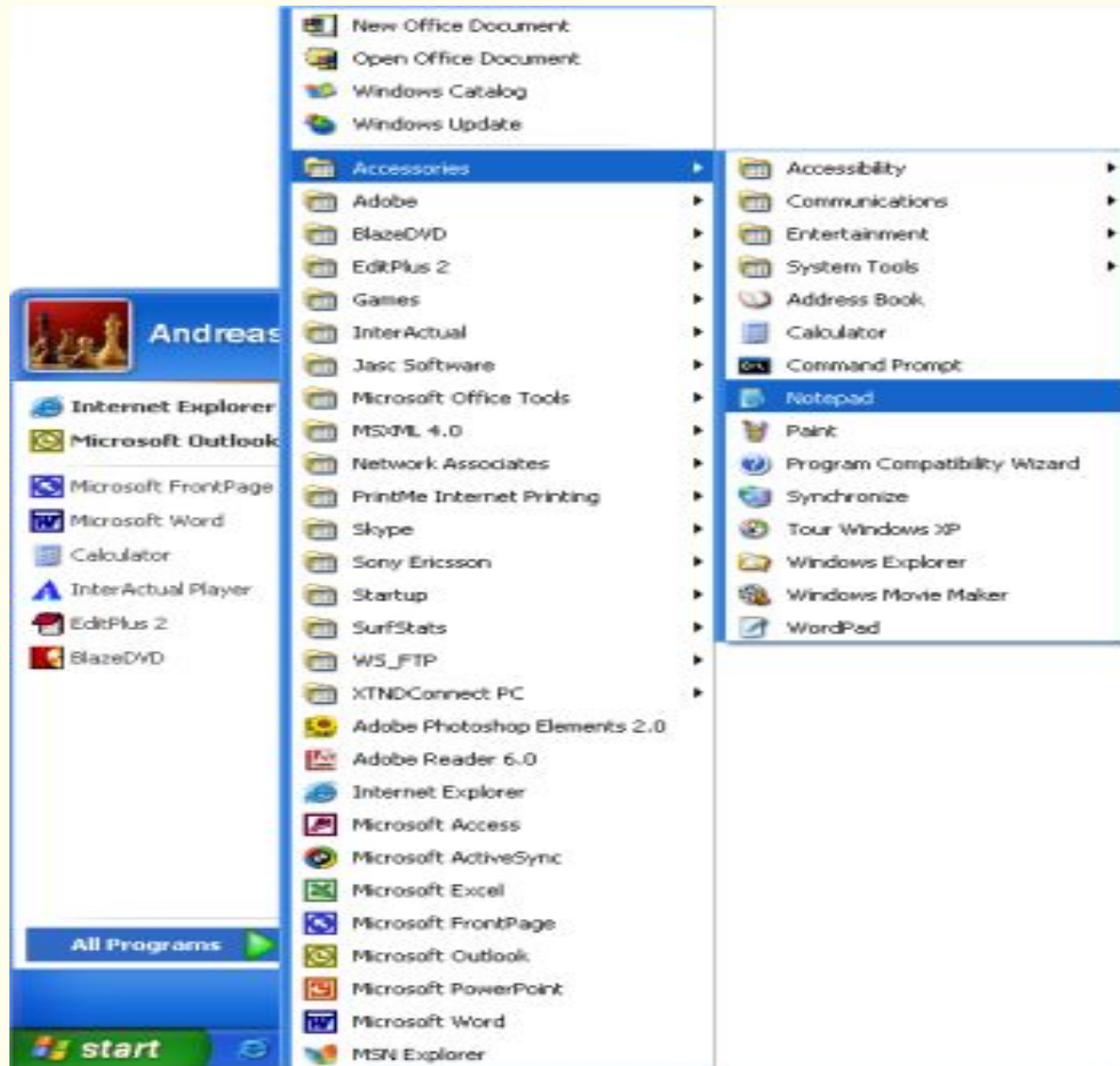
\*Если вы перейдете к следующему уроку, то уже через 10 минут сможете создать свой первый web-сайт.

# Урок 4: Создайте свой первый сайт

---

После полученных в предыдущем уроке знаний вы уже через несколько минут создадите свой первый web-сайт.





## Как?

Уроке 1 мы узнали, что необходимо для создания web-сайта: браузер и Notepad (или аналогичная программа - текстовый редактор). Поскольку вы читаете этот текст, то ваш браузер уже открыт. Вам нужно открыть лишь дополнительное окно браузера (открыть браузер ещё раз), и вы сможете читать этот учебник и одновременно видеть ваш новый web-сайт.

Также откройте Notepad (в Accessories в Programs - Start menu):

Теперь мы  
ГОТОВЫ!

# Что я могу?

- Начнём с чего-нибудь попроще. Как насчёт страницы, на которой написано: "Ура! Это моя первая web-страница." Читайте дальше, и вы узнаете, как это легко сделать.
- HTML прост и логичен. Браузер читает HTML так, как читаете его вы: сверху вниз и слева направо. Таким образом, HTML-документ начинается и заканчивается тем, чем должен начинаться и заканчиваться текст.
- Во-первых необходимо сообщить браузеру, что вы будете "говорить" с ним на языке HTML. Это делается тэгом `<html>` (тут ничего нового). Так что, прежде чем напечатать что-либо, нужно поставить тэг "`<html>`" в первой строке документа Notepad.
- Как вы, возможно, помните из предыдущего урока, `<html>` это открывающий тэг, который должен иметь и закрывающий тэг после того, как вы закончите ввод HTML. Поэтому, чтобы не забыть, поставьте закрывающий тэг "`</html>`" на пару строк ниже и вводите весь текст документа между `<html>` и `</html>`.

- Следующее, что необходимо, это "head" (голова или "шапка"), которая содержит служебную информацию о вашем документе, и "body" (тело), содержимое самого документа. Поскольку HTML логичен, head (<head> и </head>) находятся выше body (<body> и </body>).
- Ваш документ теперь должен выглядеть так:
- `<html> <head> </head> <body> </body> </html>`
- Обратите внимание, как мы структурировали тэги по строкам (с помощью клавиши Enter) и какие сделали отступы (клавишей Tab). В принципе не важно, как вы структурируете ваш HTML-документ. Но, для облегчения чтения кода, **настоятельно рекомендуем структурировать ваш HTML с помощью перевода строк и отступов**, как в нашем примере.
- Вот вы уже и сделали вашу первую web-страницу - первенца, который, возможно, выглядит не так, как вы предполагали, когда начинали читать этот учебник, но это уже, в некотором роде, web-сайт. Вы создали базовый шаблон для всех последующих HTML-документ.

# Ну ладно, но как мне добавить содержимое в мой web-сайт?

- Как вы уже видели ранее, в вашем HTML-документе есть две части: a head и body. В разделе head вы вносите информацию о странице, а body содержит информацию самой страницы.
- Например, если вы хотите дать название документу, чтобы оно появлялось в верхней строке браузера, это нужно сделать в разделе "head". Тэг для названия - <title>:
  - <title>Моя первая web-страница</title>
  - 
  -
- Обратите внимание, что этот title не появляется на самой web-странице. Всё, что вы хотите видеть на самой странице, является содержимым/content и должно, следовательно, находиться между тэгами "body".

- Итак, мы хотим, чтобы на странице выводилось "Ура! Это моя первая страница." Этот текст, который вы хотите показать, и, естественно, он должен располагаться в разделе body. Поэтому в разделе body напечатайте следующее:
  - `<p>Ура! Это моя первая страница.</p>`
  - 
  - 
  - Буква p в `<p>` это сокращение от "paragraph" и означает именно это - параграф текста.

- Ваш HTML-документ должен теперь иметь такой вид:
- `<html> <head> <title>My first page </title> </head>  
<body> <p>Hurrah! This is my first page.</p>  
</body> </html>`

Готово! Вы создали свой первый настоящий web-сайт!



Теперь нужно сохранить его на жёсткий диск, а затем открыть в вашем браузере:

- В Notepad выберите "Save as..." в меню "File".
- Выберите "All Files" в боксе "Save as type". Это очень важно - иначе вы сохраните файл как обычный текстовый документ, а не как HTML-документ.
- Теперь сохраните ваш документ как "page1.htm" (расширение ".htm" указывает, что это HTML-документ. ".html" даст тот же результат. Я всегда использую ".htm", но вы можете выбрать любое из этих двух расширений). Где вы сохраните документ на жёстком диске - не имеет значения, главное запомнить это место, чтобы потом легко найти файлы.

# Save As



Save in:

html



My Recent Documents



Desktop



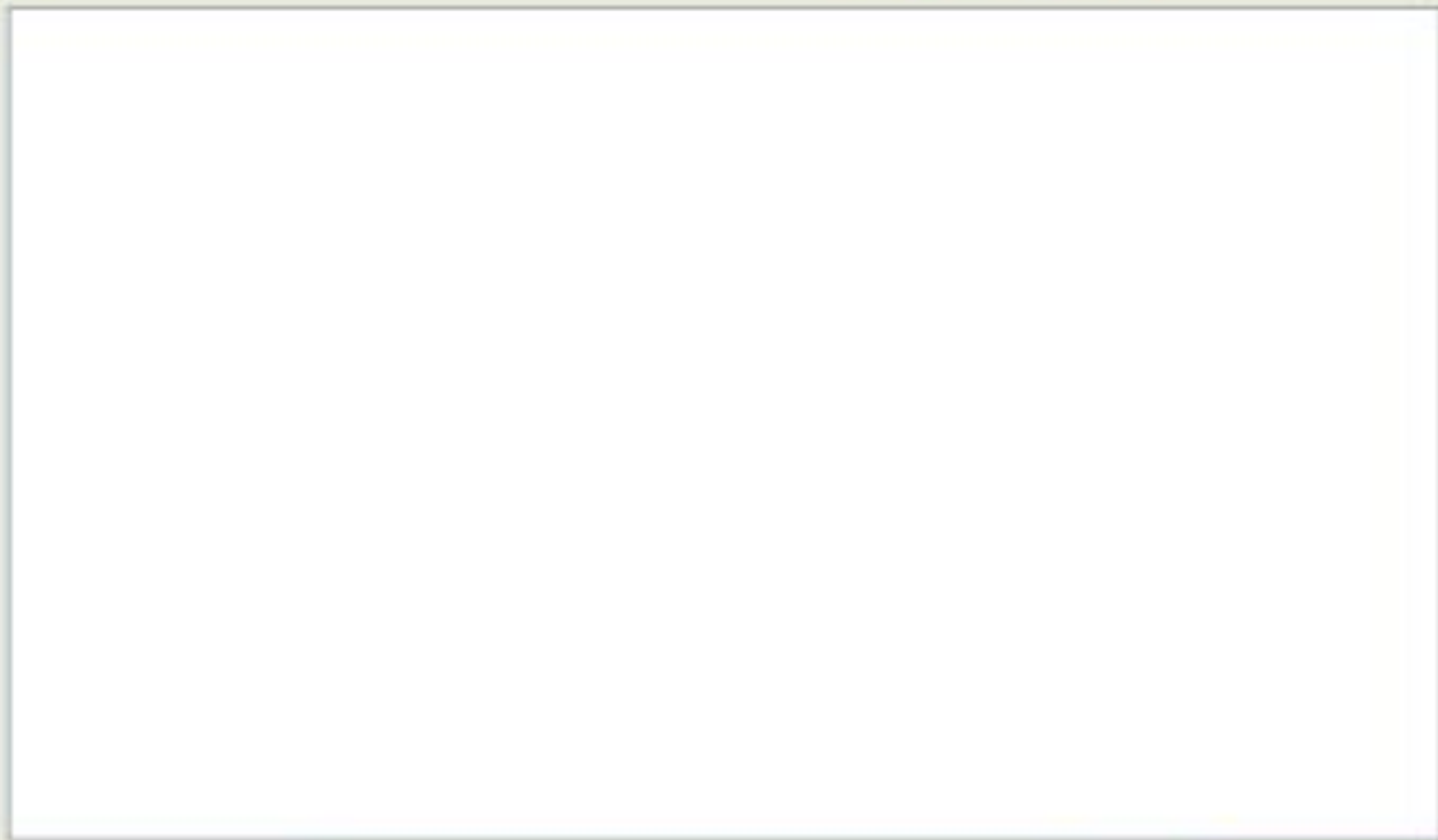
My Documents



My Computer



My Network



File name:

page1.html



Save as type:

All Files



Encoding:

ANSI

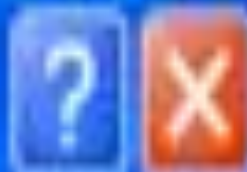


Save

Cancel

- Теперь откроем браузер:
- Выберите "Open" в меню "File" (CTRL+O).
- Щёлкните "Browse" в появившемся диалоге.
- Теперь найдите ваш HTML-документ и щёлкните "Open"

# Open



Type the Internet address of a document or folder, and Internet Explorer will open it for you.

Open:

c:\html\page1.htm

Open as Web Folder

OK

Cancel

Browse...

- Теперь вы должны увидеть страницу с текстом "Ура! Это моя первая страница." в вашем браузере. Поздравляем!

Если вы хотите, чтобы весь мир увидел ваше творение, можете перейти в Урок 14 и вы узнаете, как выгрузить вашу страницу в Internet. Или наберитесь терпения и читайте дальше. Всё ещё только начинается.

*Всегда начинайте работу с базового шаблона, который мы создали  
на предыдущем уроке.*


---

**УРОК 5: ЧТО МЫ УЖЕ  
ЗНАЕМ?**



В разделе head всегда пишете:  
<title>Название вашей  
страницы</title>. Обратите  
внимание, что title выводится в  
строке заголовка окна браузера:





❖ Этот title очень важен, поскольку используется поисковыми машинами (такими, как Google) для индексирования вашего веб-сайта и показа в результатах поиска.



Web [Images](#) [Groups](#) [News](#) [more »](#)

"Lesson 5: What have you learned so far?"

Search

[Advanced Search](#)  
[Preferences](#)

## Web

**Tip:** Try removing quotes from your search to get more results.

[What have you learned so far? - Lesson 5 | HTML Tutorial | HTML.net](#)

**Lesson 5: What have you learned so far?** Always start with the basic template we made in the previous lesson.. `<html> <head> <title></title> </head> <body> ...`

[www.html.net/tutorials/html/lesson5.asp](#) - 5k - [Cached](#) - [Similar pages](#)

❖ В разделе body вы записываете содержимое страницы. Теперь вы уже знаете несколько важных тэгов:

---

❖ `<p>Параграф.</p>`

❖ `<b>Текст жирным шрифтом.</b>`

❖ `<h1>Заголовок</h1>`

❖ `<h2>Подзаголовок</h2>`

❖ `<h3>Под-подзаголовок</h3>`

❖

❖

❖ Помните: единственный метод выучить HTML - метод проб и ошибок. Но не сомневайтесь, вы никогда не сможете повредить компьютер или Internet. Поэтому экспериментируйте - это лучший способ набраться опыта.

# ЧТО ЭТО ОЗНАЧАЕТ?

❖ Никто не сможет стать хорошим создателем web-сайтов, изучая примеры в этом учебнике. Здесь вы лишь научитесь понимать основные строительные блоки - чтобы освоить всё, вы должны использовать эти блоки по-новому и творчески.

❖ Итак, вы смело нырнули в омут с головой, крепко стоите на ногах... Окаж, может быть и нет. Но беритесь за дело и экспериментируйте с тем материалом, который изучаете.

# И ЧТО ТЕПЕРЬ?

❖ Попробуйте самостоятельно создать несколько страниц.

Например, сделайте страницу с title, heading, текстом, подзаголовком и ещё каким-нибудь текстом. Неплохо подглядывать в учебник, когда вы делаете первые страницы. Но впоследствии пытайтесь делать уже по памяти, не заглядывая в справочники.

# УРОК 6: ЕЩЁ НЕМНОГО HTML - ТЭГОВ

# ТЕПЕРЬ ЧТО?

Теперь выучим семь новых тэгов.

Таким же образом, как вы делали текст жирным шрифтом с помощью тэга `<b>`, вы можете придать ему характер курсива/italic тэгом `<i>`. И конечно, "i" это сокращение от "italic". Пример 1:

`<i>`Это должно быть выведено курсивом.`</i>`

будет выглядеть в браузере:

*Это должно быть выведено курсивом.*

# АНАЛОГИЧНО МОЖНО УМЕНЬШИТЬ РАЗМЕР ШРИФТА ТЭГОМ <SMALL>:

Пример 2:

`<small>Это будет выведено уменьшенным шрифтом.</small>`

будет выглядеть в браузере:

Это будет выведено уменьшенным шрифтом.





# МОГУ ЛИ ИСПОЛЬЗОВАТЬ НЕСКОЛЬКО ТЭГОВ ОДНОВРЕМЕННО?

Да, исключая перекрывание тэгов. Это проще увидеть на примере:

Пример 3:

Если вы хотите вывести текст **bold** и *italic*, это нужно сделать так:

```
<b><i>Текст bold и italic</i></b>
```

а НЕ так :

```
<b><i>Текст bold и italic</b></i>
```

Разница в том, что, в первом случае, мы закрыли первый тэг в последнюю очередь. Так мы избегаем конфликтов в браузере.

# ЕЩЁ ТЭГОВ!

Как сказано в Уроке 4, есть тэги, которые являются одновременно открывающими и закрывающими. Эти тэги содержат команды, которые не связаны с конкретными буквами, являются изолированными командами. Пример - тэг `<br />`, который создаёт перевод строки:

Пример 4:

Некоторый текст`<br />` и ещё текст, уже на новой строке  
будет выглядеть в браузере:

Некоторый текст

и ещё текст, уже на новой строке

Заметьте, что этот тэг записан как гибрид открывающего и закрывающего тэгов с пробелом и слэшем: `<br />`. В принципе, его можно записывать `<br></br>`, но зачем усложнять?

Другой такой тэг - `<hr />` - рисует горизонтальную линию ("`hr`" от "`horizontal rule`"):

Пример 5:

`<hr />`

будет выглядеть в браузере:



Примеры тэгов, требующих наличия и закрывающего тэга: `<ul>`, `<ol>` и `<li>`. Эти тэги используются для вывода списков.

**<ul>** - сокращение от "unordered list/неупорядоченный список" - вставляет значок кнопки для каждого элемента списка. **<ol>** - сокращение от "ordered list/упорядоченный список" - нумерует каждый элемент списка. Для создания элемента списка используйте тэг **<li>** ("list item/элемент списка"). Непонятно? См. примеры:

**Пример 7:**

**<ul> <li>Элемент списка</li> <li>Другой элемент списка</li> </ul>**

**выглядит в браузере:**

- **Элемент списка**
- **Другой элемент списка**

**Пример 8:**

```
<ol> <li>Первый элемент списка</li> <li>Второй элемент списка</li>  
</ol>
```

**выглядит в браузере:**

- 1. Первый элемент списка**
- 2. Второй элемент списка**

# ФИ! И ЭТО ВСЁ?

Всё на сегодня. Итак, экспериментируйте и создавайте ваши собственные страницы с семью новыми тэгами, изученными на этом уроке:

`<i>Italic</i>` `<small>Уменьшенный шрифт</small>` `<br />` перевод строки  
`<hr />` Горизонтальная линия `<ul>Список</ul>` `<ol>Упорядоченный список</ol>` `<li>Элемент списка</li>`

# Урок 7: Атрибуты

Вы можете назначать атрибуты во многих тэгах.

# Что такое атрибут?

- 0 Как вы, возможно, помните, тэги это метки, указывающие браузеру, как нужно представлять ваш web-сайт. (Например, `<br />` информирует браузер, что нужно сделать перевод строки). В некоторых тэгах вы можете вводить дополнительную информацию. Такая дополнительная информация называется "атрибут".
- 0 **Пример 1:**
- 0 `<h2 style="background-color:#ff0000;">Мой друг HTML</h2>`
- 0 Атрибуты всегда записываются внутри тэга, после них следует знак равенства и детали атрибута, заключённые в двойные кавычки. Точка с запятой после атрибута служит для разделения команд разных стилей. Мы к этому ещё вернёмся.



# О чём же речь?

- 0 Атрибуты бывают разные. Первый изученный вами атрибут это `style/стиль`. **С помощью этого атрибута вы можете настраивать отображение вашего web-сайта.** Например, цвет фона/`background colour`:
- 0 **Пример 2:**
- 0 `<html> <head> </head> <body style="background-color:#ff0000;" </body> </html>`
- 0 выведет в браузере страницу красного цвета - проверьте сами. Мы детально объясним, как работает цветовая система.
- 0 Обратите внимание, что некоторые тэги и атрибуты используют американскую разновидность английского языка, т. е. `color` вместо `colour`. Важно использовать именно такой вариант написания, как мы используем в примерах в этом учебнике - иначе браузеры не смогут понять ваш код. Также не забывайте ставить двойные кавычки после атрибута.

# Как страница стала красной?

- o В предыдущем примере мы запросили вывод красного фона кодом "#ff0000". Это код красного цвета в т. н. шестнадцатиричном формате (HEX). **Каждому цвету соответствует своё 16-ричное число.** Вот примеры:
- o Белый: #ffffff  
Чёрный: #000000 (нули)  
Красный: #ff0000  
Синий: #0000ff  
Зелёный: #00ff00  
Жёлтый: #ffff00
- o 16-ричный код цвета состоит из # и шести цифр или букв. Существует более 1000 HEX кодов и бывает непросто определить, какой HEX-код какому цвету соответствует. Для удобства мы советуем иметь карту 216 самых распространённых кодов цвета: [216 Web Safe Colour Chart](#).
- o [http://ru.html.net/tutorials/html/lesson7\\_216websafecolourchart.php](http://ru.html.net/tutorials/html/lesson7_216websafecolourchart.php)
- o Можно использовать также английские названия самых распространённых цветов (white, black, red, blue, green и yellow).
- o **Пример 3:**
- o `<body style="background-color: red;">`
- o Но хватит о цвете. Давайте вернёмся к атрибутам.

# Какие тэги могут использовать атрибуты?

- 0 **Различные атрибуты могут применяться для большинства тэгов.**
- 0 В таких тэгах, как `body`, вы будете часто использовать атрибуты, а, например, в тэге `br` - редко, поскольку перенос строки это обычно перенос строки без каких-либо уточняющих параметров.
- 0 Имеется множество тэгов, есть и много атрибутов. Есть атрибуты, предназначенные специально для какого-то определённого тэга, а другие можно использовать в разных тэгах. И наоборот: некоторые тэги могут иметь только один какой-то определённый атрибуты, в то время как другие тэги - несколько атрибутов.
- 0 Это может звучать немного непонятно, но, когда вы выучите различные атрибуты, вы увидите, что всё построено очень логично, что использовать атрибуты легко и просто, и что они значительно расширяют ваши возможности.
- 0 В этом учебнике мы рассмотрим самые важные атрибуты.

# Кстати, из каких ещё частей состоит тэг?

- Основная часть тэга называется элемент (например, `p` в `<p>`). Таким образом, тэг состоит из элемента (например, `<p>`), либо из элемента плюс один или более атрибутов (например, `<p style="background-color:#ff0000;">`). Проще некуда.

## Урок 8: Ссылки

В этом уроке вы научитесь создавать ссылки - переходы с одной страницы на другую.

# Что необходимо для создания ссылки?

- Для создания ссылки вы используете то же, что и всегда при кодировании HTML: тэг. **Простой тэг с одним элементом и одним атрибутом - и вы можете перейти куда угодно.** Вот пример того, как может выглядеть ссылка на HTML.net:
- Пример 1:
- `<a href="http://www.html.net/">Это ссылка на HTML.net</a>`
- будет выглядеть в браузере:
- [Это ссылка на HTML.net](http://www.html.net/)

- Элемент `a` обозначает "якорь/anchor". Атрибут `href` это сокращение от "hypertext reference/гипертекстовая ссылка", специфицирующий место, на которое выполняется переход по данной ссылке - обычно это internet-адрес и/или имя файла.
- В примере атрибут `href` имеет значение "`http://www.html.net`", которое является полным адресом HTML.net и называется URL (Uniform Resource Locator/универсальный локатор ресурса). Обратите внимание, что "`http://`" всегда должно входить в состав URLов. Слова "Это ссылка на HTML.net" это текст, который показывается в браузере как ссылка. Не забудьте закрыть тэг `</a>`.

# Как насчёт ссылок между моими собственными страницами?

- Если вы делаете ссылки между страницами на одном web-сайте, то не нужно указывать полный адрес (URL) документа. Например, если у вас две страницы (назовём их `page1.htm` и `page2.htm`), сохранённые в одной папке, вы можете сделать ссылку с одной страницы на другую, просто напечатав имя файла в ссылке. То есть ссылка с `page1.htm` на `page2.htm` будет выглядеть так:
  - Пример 2:
    - `<a href="page2.htm">Щёлкните здесь для перехода на page 2</a>`
    - Если `page 2` помещена в подпапку "`subfolder`", ссылка выглядит так:
      - Пример 3:
        - `<a href="subfolder/page2.htm">Щёлкните здесь для перехода на page 2</a>`
        - В обратную сторону ссылка со страницы `page 2` (в подпапке `subfolder`) на `page 1` будет такой:
          - Пример 4:
            - `<a href="../page1.htm">Ссылка на page 1</a>`
            - Сочетание `"../"` указывает на папку, расположенную на один уровень выше от данной позиции файла, с которого делается ссылка. Следуя этой логике, вы можете также указать на два уровня выше `"../../"` или более.
- Разобрались? Альтернативно вы всегда можете указать полный адрес файла (URL).



# А ссылки внутри страницы?

- Вы можете также создавать ссылки-переходы внутри самой страницы - например, оглавление со ссылками на главы. Всё, что вам необходимо, - использовать атрибут `id` и символ "#".
- Используйте атрибут `id` для маркировки элемента, на который вы хотите сделать переход. Например:
  - `<h1 id="heading1">heading 1</h1>`
  - Теперь можно создать ссылку на этот элемент с помощью знака "#" в атрибуте ссылки. Знак "#" сообщает браузеру, что это переход на той же самой странице. После "#" должен следовать `id` тэга, на который выполняется переход. Например:
    - `<a href="#heading1">Ссылка на heading 1</a>`

- Всё станет понятнее на примере:
- Пример 5:
- ```
<html> <head> </head> <body> <p><a href="#heading1">Ссылка на heading 1</a></p> <p><a href="#heading2">Ссылка на heading 2</a></p> <h1 id="heading1">heading 1</h1> <p>Text text text text</p> <h1 id="heading2">heading 2</h1> <p>Text text text text</p> </body> </html>
```
- выглядит в браузере:
- [Ссылка на heading 1](#)
- [Ссылка на heading 2](#)
- Heading 1
- Text text text text
- Heading 2
- Text text text text
- (Примечание: атрибут id должен начинаться с буквы)

# Могу я перейти ещё куда-нибудь?

- Вы можете также сделать ссылку на e-mail адрес почти так же, как в ссылке на документ:
- Пример 6:
- Ссылка на `<a href="mailto:nobody@html.net">Отправить e-mail nobody на HTML.net</a>`
- будет в браузере:
- [Отправить e-mail nobody на HTML.net](mailto:nobody@html.net)
- Единственное отличие в ссылках на e-mail и на файл состоит в том, что, вместо адреса документа, вы записываете mailto: с последующим адресом электронной почты/e-mail. При щелчке по ссылке открывается программа по умолчанию для работы с электронной почтой с адресом, уже записанным в строке адреса. Обратите внимание, что эта функция будет работать только в том случае, если e-mail программа установлена на вашем компьютере. Попробуйте!

# Есть ещё какие-нибудь другие атрибуты, которые мне нужно знать?

- Для создания ссылки вы обязательно должны использовать атрибут href. Кроме того, в ссылку можно поместить title:
- Пример 7:
- `<a href="http://www.html.net/" title="Посещайте HTML.net и изучайте HTML">HTML.net</a>`
- будет выглядеть в браузере:
- [HTML.net](http://www.html.net/)
- Атрибут title используется для краткого описания ссылки. Если вы - не щёлкая мышью - поместите её указатель над ссылкой, вы увидите, как появится текст "Посещайте HTML.net и изучайте HTML".

Здóрово было бы поместить портрет актёра и музыкальной легенды David'a Hasselhoff'a прямо в центре вашей страницы?!

Урок 9:

Изображения



- » Может быть, но это довольно просто сделать. Всё, что вам необходимо, как всегда, - тэг:
- » **Пример 1:**
- » ``
- » будет выглядеть в браузере:

Это звучит  
немного  
дерзко...





- » Вам необходимо сообщить браузеру, что вы хотите вставить изображение/image (img), и указать его размещение (src, сокращение для "source"). У вас есть файл изображения?
- » Обратите внимание, что тэг img не требует наличия закрывающего тэга. Как и <br />, это команда не связана с буквенным текстом.
- » "david.jpg" это название файла изображения. ".jpg" - расширения файла, тип изображения. Как ".htm" указывает, что файл является HTML-документом, так и ".jpg" сообщает браузеру, что файл является изображением. Есть три типа файлов изображений, которые можно вставить на ваши страницы:
  - » GIF (Graphics Interchange Format)
  - » JPG / JPEG (Joint Photographic Experts Group)
  - » PNG (Portable Network Graphics)
  - » GIF-изображения обычно лучше всего для графики и рисунков, а JPEG - для фотографий. Для этого есть две причины: первая - GIF-изображения содержат только до 256 цветов, а JPEG - до миллионов цветов, и второе - формат GIF лучше сжимает простые изображения, чем JPEG, который оптимизирован под более сложные изображения. Чем выше сжатие, тем меньше файл изображения, тем быстрее загружается ваша страница. Как вы, вероятно, знаете из собственного опыта, пользователи не любят "тяжёлые" страницы.
- » Традиционно форматы GIF и JPEG являются преобладающими типами, но в последнее время становится всё более популярным формат PNG (прежде всего из-за издержек формата GIF). Формат PNG по многим параметрам превосходит JPEG и GIF: миллионы цветов и эффективное сжатие.



- » Для этого вам понадобится программа редактирования файлов изображений. Такая программа совершенно необходима для создания интересных web-сайтов.

Где мне взять  
файлы  
изображений? >

- » К сожалению в Windows и других операционных системах отсутствуют хорошие программы для редактирования изображений. Таким образом, вы можете вложить деньги в приобретение Paint Shop Pro, PhotoShop или Macromedia Fireworks - это три лучшие программы редактирования изображений, имеющиеся в данный момент на рынке.
- » Однако, как мы говорили ранее, не обязательно покупать дорогие программы для работы с этим учебником. Вы можете загрузить прекрасную программу для работы с изображениями, Irfan View, которая является т. н. freeware, и, следовательно, ничего не стоит.
- » Либо вы можете просто содрать изображения с других сайтов, загрузив соответствующие страницы. Но, пожалуйста, будьте внимательны, чтобы не нарушить при этом авторских прав. Вот как загрузить изображения:
  - » Правой клавишей мыши щёлкните на изображении в Internet.
  - » Выберите "Save picture as..." в меню.
  - » Выберите место для сохранения на вашем компьютере и нажмите "Save".
  - » Прочитайте это с изображением, расположенным здесь, и сохраните его на вашем компьютере в том же месте, что и ваши HTML-документы. (Заметьте, что этот логотип сохраняется в файле формата PNG: logo.png):



HTML5.net

Теперь вы можете  
вставлять это  
изображение в  
ваши страницы.  
Попробуйте  
сделать это  
самостоятельно.



Это всё, что  
нужно знать об  
изображениях? >

- » Вам необходимо знать ещё две вещи.
- » Во-первых, вы можете легко вставлять изображения, размещённые в других папках, или даже на других web-сайтах:
- » **Пример 2:**
- » ``
- » **Пример 3:**
- » ``
- » Во-вторых, изображения могут быть ссылками:
- » **Пример 4:**
- » `<a href="http://www.html.net"> </a>`
- » будет выглядеть в браузер примерно так (попробуйте щёлкнуть на изображении):





HTML.net

Есть ещё  
атрибуты,  
которые мне  
необходимы?





- » Вам всегда нужно использовать атрибут `src`, который указывает браузеру, где находится изображение. Помимо этого есть ещё и другие атрибуты, которые могут понадобиться при работе с изображениями.
- » Атрибут `alt` используется для альтернативного описания изображения, если, по каким-то причинам, оно не показано пользователю. Это особенно касается пользователей с ослабленным зрением, или если страница очень медленно загружается. Следовательно, всегда нужно использовать атрибут `alt`:
- » **Пример 5:**
- » ``
- » Некоторые браузеры отображают текст этого атрибута в небольшом боксе при наведении указателя мыши на изображение. Обратите внимание, что при использовании атрибута `alt` целью является дать альтернативное описание изображения. Атрибут `alt` не следует использовать для создания всплывающих сообщений, поскольку тогда пользователи со слабым зрением будут вынуждены слушать сообщение, не зная, что изображено на картинке.
- » Атрибут `title` можно использовать для краткого описания изображения:
- » **Пример 6:**
- » ``



- » Если вы поместите курсор, без щелчка, над ссылкой, вы увидите, что текст "Learn HTML from HTML.net" появился во всплывающем боксе.
- » Два других важных атрибута - width и height:
- » **Пример 7:**
- » ``
- » даст в браузере:
- »





- » Атрибуты `width` и `height` можно использовать для установки ширины и высоты изображения, соответственно. Значение указывается в пикселах. Пиксел это единица для измерения разрешения экрана. (Обычное разрешение - 800x600 и 1024x768 пикселей). В отличие от сантиметров, пиксели являются относительными единицами, которые зависят от разрешения данного экрана. Для пользователя с высоким разрешением экрана 25 пикселей могут выглядеть как 1 сантиметр, а эти же 25 пикселей при низком разрешении могут соответствовать 1.5 сантиметрам экрана.
- » Если вы не установите `width` и `height`, изображение будет показано в своём реальном размере. При помощи `width` и `height` вы можете изменять размеры:
- » **Пример 8:**
- » ``



- » Неплохо помнить, что реальный размер файла изображения в килобайтах останется неизменным, и этот файл будет занимать то же время для загрузки, как и раньше, хотя и будет выглядеть на экране уменьшенным. Следовательно, никогда не уменьшайте размер изображения атрибутами `width` и `height`. Вместо этого уменьшайте размер изображения в программе редактирования.
- » Тем не менее, неплохо использовать атрибуты `width` и `height`, поскольку браузер сможет определять размер изображения на странице до его полной загрузки. Это позволит браузеру быстрее и более качественно загружать страницы.
- » Ну, хватит о David Hasselhoff и изображениях.



Таблицы используются, когда вам необходимо показать "табличные данные", например, информацию, логически упорядоченную в столбцы и ряды

## УРОК 10: ТАБЛИЦЫ

---

# ЭТО ТРУДНО?

---

- ▣ остроение таблиц в HTML может показаться сложным на первый взгляд, но, если вы не будете дёргаться, это окажется строго логичным - как и всё в HTML.

- ▣ Пример 1:

```
<table>
<tr>
<td>Cell 1</td>
<td>Cell 2</td>
</tr>
<tr>
<td>Cell 3</td>
<td>Cell 4</td>
</tr>
</table>
```

будет выглядеть в браузере:

- ▣ Cell 1
- ▣ Cell 3

- Cell 2
- Cell 4

# КАКАЯ РАЗНИЦА МЕЖДУ <TR> И <TD>?

- Как видно из приведённого примера, это, возможно, самый сложный из примеров HTML, которые мы вам давали до сих пор. Давайте здесь остановимся и сделаем пояснения:
- Для вставки таблиц используются 3 базовых тэга:
- `<table>` - начало и конец таблицы. Логично.
- `<tr>` - "table row/ряд таблицы", начинает и заканчивает горизонтальный ряд ячеек. Тоже логично.
- `<td>` - сокращение от "table data/табличные данные". Этот тэг начинает и заканчивает каждую ячейку ряда таблицы. Всё просто и логично.
- Вот что происходит в Примере 1: таблица начинается с `<table>`, затем идёт `<tr>`, указывающий на начало нового ряда. В ряду вставлены две ячейки: `<td>Cell 1</td>` и `<td>Cell 2</td>`. Ряд закрывается `</tr>`, и сразу начинается новый ряд `<tr>`. В новом ряду также две ячейки. Таблица закрывается `</table>`.
- Чтобы было ещё понятнее: ряды это горизонтальные строки ячеек, а столбцы - вертикальные столбцы ячеек:

<b>Cell 1</b>	<b>Cell 2</b>
<b>Cell 3</b>	<b>Cell 4</b>



- 
- Cell 1 и Cell 2 образуют ряд. Cell 1 и Cell 3 образуют столбец.
  - В вышеприведённом примере в таблице имеются два ряда и два столбца. Но в таблице может быть неограниченное число рядов и столбцов.

Пример 2:

```
<table><tr><td>Cell 1</td><td>Cell 2</td><td>Cell 3</td><td>Cell 4</td></tr><tr><td>Cell 5</td><td>Cell 6</td><td>Cell 7</td><td>Cell 8</td></tr><tr><td>Cell 9</td><td>Cell 10</td><td>Cell 11</td><td>Cell 12</td></tr></table>
```

□

□

будет выглядеть в браузере:

- Cell 1 Cell 2 Cell 3 Cell 4
- Cell 5 Cell 6 Cell 7 Cell 8
- Cell 9 Cell 10 Cell 11 Cell 12

# ЕСТЬ ЕЩЁ КАКИЕ-НИБУДЬ АТТРИБУТЫ?

---

Разумеется. Например, атрибут `border` используется для специфицирования толщины рамки вокруг таблицы:

Пример 3:

```
<table border="1">  
<tr>  
<td>Cell 1</td>  
<td>Cell 2</td>  
</tr>  
<tr>  
<td>Cell 3</td>  
<td>Cell 4</td>  
</tr>  
</table>
```

Толщина рамки специфицируется в пикселах (См. Урок 9)

Как и с изображениями, вы можете указывать размер таблицы в пикселах или в процентах от размера экрана:

Пример 4:

```
<table border="1" width="30%">
```

Этот пример будет отображён в браузере как таблица шириной в 30% экрана. Проверьте сами.

# ЕЩЁ АТТРИБУТЫ?

---

- У таблиц есть много атрибутов. Вот ещё два:
- align: специфицирует выравнивание содержимого всей таблицы по горизонтали, в ряду или в отдельной ячейке. Например, left, center или right.
- valign: специфицирует выравнивание по вертикали в ячейке. Например, top, middle или bottom.
- Пример 5:

- `<td align="right" valign="top">Cell 1</td>`
- 
-

# ЧТО МОЖНО ВСТАВЛЯТЬ В ТАБЛИЦЫ?

---

- Теоретически - всё: текст, ссылки и изображения... НО таблицы предназначены для вывода табличных данных (т. е. данных, которые изначально представляются в виде столбцов и рядов), поэтому избегайте вставки данных в таблицы только потому, что вы хотите разместить их рядом друг с другом.
- В добрые старые времена Internet - т. е. несколько лет назад - таблицы часто использовались для распределения содержимого на странице. Однако сегодня для этого есть более крутой способ - CSS. Но об этом позже.
- Теперь примените ваши знания на практике и создайте несколько таблиц различных размеров, с разными атрибутами и содержимым. Это займёт у вас несколько часов.

# Урок 11: Ещё о таблицах

Название "Ещё о таблицах" может звучать немного вызывающе. Если вы уже можете создавать таблицы, то уже абсолютно ничто в HTML не может смутить вас.

# Что же ещё?

- При создании таблиц используются два атрибута: `colspan` и `rowspan`.
- `Colspan` - сокращение от "column span/охват столбцов". **Colspan используется в тэге `<td>` для специфицирования того, сколько столбцов охватывает данная ячейка:**
- **Пример 1:**
- ```
<table border="1"> <tr> <td colspan="3">Cell 1</td>  
</tr> <tr> <td>Cell 2</td> <td>Cell 3</td> <td>Cell  
4</td> </tr> </table>
```

- Установка `colspan "3"`, заставляет ячейку в первом ряду охватывать три столбца. Если установим `colspan "2"`, ячейка охватит только два столбца, и понадобится вставить дополнительные ячейки в первый ряд, чтобы ячейки ровно распределились на два ряда.

- **Пример 2:**

- ```
<table border="1"> <tr> <td colspan="2">Cell 1</td>
<td>Cell 2</td> </tr> <tr> <td>Cell 3</td> <td>Cell
4</td> <td>Cell 5</td> </tr> </table>
```

# А как насчёт rowspan?

- Как вы уже могли догадаться, **rowspan** специфицирует, сколько **рядов охватывает** данная ячейка:
- **Пример 3:**
- ```
<table border="1"> <tr> <td rowspan="3">Cell 1</td>  
<td>Cell 2</td> </tr> <tr> <td>Cell 3</td> </tr>  
<tr> <td>Cell 4</td> </tr> </table>
```



- В этом примере `rowspan` имеет значение "3" в ячейке Cell 1. Это указывает, что ячейка должна охватывать три ряда (свой собственный ряд плюс ещё два). Cell 1 и Cell 2 при этом остаются в одном ряду, а Cell 3 и Cell 4 образуют отдельные ряды.
- Удивлены? Ладно, это не так сложно. Неплохо будет нарисовать таблицу на бумаге до начала работы в HTML.
- Неудивительно? Тогда вернитесь к началу и создайте пару таблиц с использованием `colspan` и `rowspan`.

# УРОК 12: ПУБЛИКАЦИЯ СТРАНИЦ

До сих пор только вы имели возможность наслаждаться просмотром ваших страниц. Теперь пришло время и всем остальным оценить ваши произведения.

# Готов ли мир к этому?

- Мир готов - скоро и вы тоже будете готовы. Для показа вашего web-сайта в Internet вы должны иметь пространство на сервере и бесплатную FTP-программу.
- Если у вас есть доступ в Internet, у вас, возможно, уже есть немного места для вашего web-сайта. Ваше пространство на сервере может называться примерно так - `http://home.provider.com/~usernumber`. Но сначала вы должны активировать его. Прочтите об этом в документации вашего Internet-провайдера или на его справочных страницах.
- Другая возможность получить некоторое пространство в Internet - настроить учётную запись e-mail (на, например, Hotmail), тогда вы сможете зарегистрировать свободное пространство в Internet. Такой сервис имеют несколько компаний, среди них - `ooowebhost.com` (щёлкните "Order" и выберите free membership) - регистрация займёт всего несколько минут.
- Для доступа на сервер вам необходимо знать "Host Name" (Например, `ftp.htmlnet.site50.net`) и иметь своё username и password.

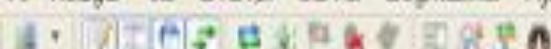
# Это всё, что мне нужно?

---

- Для доступа на сервер и выгрузки на него ваших страниц вам понадобится также программа, работающая по протоколу FTP. Возможно, у вас её ещё нет, но можно загрузить её бесплатно.
- Имеется множество FTP-программ. Одна из лучших - FileZilla, абсолютно бесплатная. Можно скачать FileZilla с [filezilla.sourceforge.net](http://filezilla.sourceforge.net).

# А как выгружать страницы?

- Ниже мы даём описание того, как выгружать страницы на бесплатную учётную запись/account на [ooowebhost.com](http://ooowebhost.com) с помощью FileZilla. Но эта процедура применима, в большей или меньшей степени, для всех провайдеров и FTP-программ.
- Откройте FTP-программу после того, как подключитесь к Internet. Вставьте "Host Name" ("[ftp.htmlnet.site50.net](http://ftp.htmlnet.site50.net)" ниже "Address"), имя пользователя (ниже "User") и пароль (ниже "Password") и щёлкните "Connect". У вас теперь имеется доступ к этому серверу. В одной части программы вы видите содержимое вашего компьютера ("Local Site"), а в другой - содержимое сервера ("Remote Site"):


 Vært: ftp.htmlnet.site50.net Brugernavn: a7146500 Password: \*\*\*\*\* Port:  Lyttelst: 

 Kommando: PLS0  
 Respons: 130 Accepted data connection  
 Respons: 226-Options: -a -l  
 Respons: 226 Transfer total  
 Respons: Happendhold vist  
 Respons: 421 Timeout - try typing a little faster next time  
 Fejl: Forbindelse til server afbrudt: CONNECTION ABORTED - Connection aborted

Lokal stier: C:\Documents and Settings\Andreas detrup\Skrivebord\HTML

Fjernst: /pubk\_html

- 📁 Skrivebord
  - 📁 Downloads
  - 📁 dvdram-win&-5025-en
  - 📁 forside
  - 📁 HTML
  - 📁 images

- 📁 /
  - 📁 public\_html

Filnavn /	Filstør...	Filtype	Sidst ændret
..			
index.html	316	HTML-dok...	22-10-2011 1...

Filnavn /	Filstør...	Filtype	Sidst ændret	Tilgæbel	Ejer
..					
htaccess	91	HTACC...	22-10-201...	0644	714
default.php	9.062	PHP-f	22-10-201...	0644	714
index.html	302	HTML-d...	22-10-201...	0644	714

Valgt 1 fil. Samlet størrelse: 316 byte

3 filer. Samlet størrelse: 8.455 byte

Server/Lokal fil	Retn...	Fl på fjernsted	Størrelse	Prior...	Status
------------------	---------	-----------------	-----------	----------	--------

- Найдите ваши HTML-документы и изображения на вашем компьютере (в "Local site") и перешлите их на сервер ("Remote site") дважды щёлкнув на них. Теперь их видит весь мир! (Например, по адресу <http://htmlnet.site50.net/page1.htm>).
- Назовите одну из страниц "index.htm" (или "index.html"), и она автоматически станет стартовой страницей. т. е., если вы введёте <http://htmlnet.site50.net> (без указания имени файла), откроется <http://htmlnet.site50.net/index.htm>.
- Для длительной работы неплохо приобрести собственный домен ([www.ваше-имя.com](http://www.ваше-имя.com)) и избавиться от длинного адреса, предоставляемого Internet-провайдером бесплатного сервиса. Можете купить домен, к примеру, на Speednames или NetworkSolutions.

# Урок 13: Web-стандарты и проверка

*В этом уроке мы дадим несколько  
больше теории HTML.*



# Что ещё нужно знать о HTML?

HTML можно кодировать несколькими способами. Браузеры также могут читать HTML по-разному. Можно сказать, что в HTML есть много диалектов. Поэтому сайты выглядят по-разному в различных браузерах.

Почти с момента зарождения Internet начались попытки выработать общий HTML-стандарт на [World Wide Web Consortium \(W3C\)](#), который основал Tim Berners-Lee (ага! тот крутой парень, который изобрёл HTML). Но это длинный и долгий путь.

В старину - когда за браузеры надо было заплатить - Netscape был основным браузером. Существовали HTML-стандарты .0 и 3.2. Но на рынке, на 90% занятом Netscape не было необходимости в общих стандартах. Кроме того, Netscape изобретал собственные странные элементы, которые не работали на других браузерах.

Многие годы Microsoft почти игнорировал Internet. Затем началось соревнование с Netscape, и появился новый браузер. Первые версии нового браузера Microsoft'a - Internet Explorer - поддерживали HTML-стандарты не лучше, чем Netscape. Но Microsoft решил распространять свой браузер бесплатно (это всегда приветствуется), и Internet Explorer вскоре стал самым популярным браузером.

Начиная с версий 4 и 5, Microsoft всё более и более поддерживал HTML-стандарты от W3C. Netscape не занимался разработкой новых версий и продолжал выпускать устаревшую версию 4.

Конец этой истории. Сегодня HTML-стандарты называются 4.01 и XHTML. Теперь уже Internet Explorer занимает свыше 90% рынка. В Internet Explorer тоже есть свои необычные элементы, но он также поддерживает и W3C HTML-стандарты. Так же делают и другие все браузеры, такие как Mozilla, Opera и Netscape.

**Итак, если вы кодируете HTML, придерживаясь стандартов W3C, вы делаете web-сайты, которые читаются всеми браузерами - сейчас и в будущем. И, какая удача, то, что вы изучили в этом учебнике, это новая и более строгая версия HTML, которая называется XHTML.**

# Круто! Можно, я расскажу ещё кому-нибудь?

При наличии различных разновидностей HTML вы должны сообщать браузеру, на каком "диалекте" HTML говорите вы, в нашем случае - XHTML. Для этого используйте Document Type Definition/определение типа документа. DTD всегда записывается в самом начале документа:

## Пример 1:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"> <html xmlns="http://www.w3.org/1999/xhtml"
lang="en"> <head> <title>Title</title> </head> <body> <p>text text</p> </body> </html>
```

Помимо Document Type Definition (первая строка пример), которое сообщает браузеру, что вы пишете на языке XHTML, вы должны вставить также дополнительную информацию в тэг html с помощью атрибутов xmlns и lang.

xmlns, это сокращение от "XML-Name-Space", всегда должен иметь значение <http://www.w3.org/1999/xhtml>. Это всё, что нужно знать. Но, если вы интересуетесь сложными вопросами, можете прочесть о namespaces на [W3C web-сайте](#).

В атрибуте lang вы указываете, на каком языке написан документ. Для этого используется стандарт [ISO 639](#), в котором перечислены коды всех языков мира. В предыдущем примере установлен English ("en").

С помощью DTD браузер точно знает, как он должен читать и отображать ваш HTML. Используйте этот образец как шаблон для ваших будущих HTML-документов.

DTD также имеет важное значение при проверке ваших страниц.


# Проверке? Почему и как я должен это делать?

Вставьте DTD в ваши страницы - и ваш HTML всегда можно будет проверить на наличие ошибок с помощью бесплатного проверщика W3C's free validator.

Чтобы протестировать эту возможность, создайте страницу и поместите её в Internet. Теперь перейдите на [validator.w3.org](http://validator.w3.org), введите адрес (URL) вашей страницы и проверьте её. Если ваш HTML корректен, вы получите congratulations message. В ином случае - сообщение об ошибках с точным указанием того, что и где сделано неправильно. Сделайте специально несколько ошибок и посмотрите, что получится.

Этот validator полезен не только для выявления ошибок. Некоторые браузеры пытаются автоматически исправлять ошибки кодировщиков и отображают страницу так, как она должна выглядеть (по их "усмотрению"). При этом вы можете никогда не увидеть ошибок в своём браузер. Однако браузеры корректируют ошибки по-разному или могут вообще не показать страницу. Проверщик-validator помогает выявить ошибки, о наличии которых вы, может быть, и не подозревали.

Всегда проверяйте ваши страницы для обеспечения их корректности.

A stylized, monochromatic illustration of a plant with several large, pointed leaves and a cluster of small, round buds or flowers on a stem, set against a dark brown background on the left side of the slide.

Поздравляем, вы добрались до  
последнего урока.

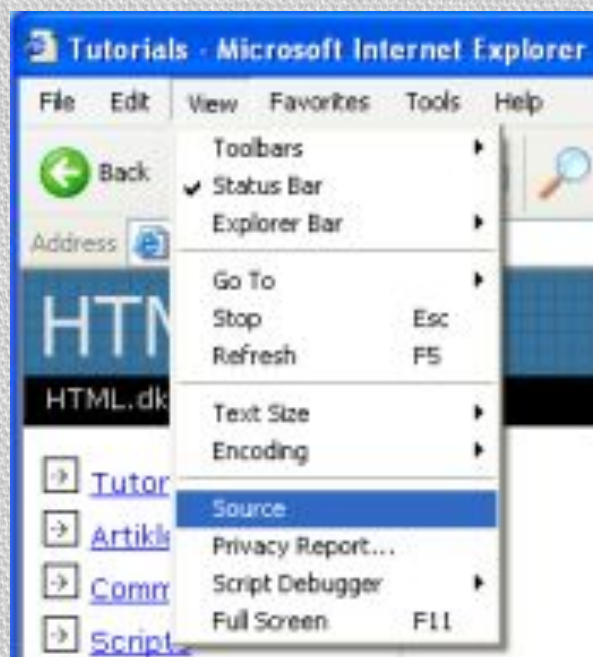
## УРОК 15: ПОСЛЕДНИЕ ЗАМЕЧАНИЯ

# Теперь я знаю всё?

- Вы много узнали и способны теперь создавать собственные web-сайты! Однако то, чему вы научились, это только основы, а есть ещё много и много чего. Но вы заложили хороший фундамент, на котором можно начинать строительство.
- В это уроке мы дадим несколько заключительных наставлений:
- Во-первых, неплохо следить за порядком и структурой ваших HTML-документов. Публикуя хорошо упорядоченные документы, вы не только демонстрируете другим уровень своего мастерства владения HTML, но и облегчаете себе работу.
- Соблюдайте стандарты и проверяйте ваши страницы. Это не слишком обременительно: всегда пишите на чистом XHTML, используйте DTD и проверяйте ваши страницы с помощью проверщика [validator.w3c.org](http://validator.w3.org).
- Делайте страницы содержательными. Помните, что HTML позволяет показывать вашу информацию в Internet, поэтому позаботьтесь, чтобы эта информация была презентабельной. Страницы могут быть очень приятными на вид, но большинство людей ищет в Internet информацию.
- Не перегружайте страницы изображениями и другими наворотами, которые вы нашли в Internet. Это замедляет загрузку страниц и мешает посетителям. Страницы, требующие для загрузки более 20 секунд, теряют до 50% посетителей.
- Не забудьте добавить ваш сайт в поисковые машины/директории, чтобы не только члены вашей семьи могли найти ваши страницы в сети. На первых страницах всех поисковых машин вы найдёте ссылки на новые страницы (самые известные машины - Google, см. также DMOZ, Yahoo, AltaVista, AlltheWeb и Lycos).
- В этом учебнике вы научились использовать Notepad, простой в использовании редактор, но, возможно, вы захотите использовать более навороченные программы, имеющие больше возможностей. Обзор таких программ есть на [Download.com](http://Download.com).

# Как узнать больше?

- Прежде всего, не прекращайте работать и экспериментировать. Изучайте web-сайты других людей и, если найдёте что-то интересное, посмотрите, как это сделано, через меню "View Source/Просмотр исходного кода" (меню "View" и "Source" вашего браузера).



- Ищите в Internet примеры и статьи о HTML. Есть множество web-сайтов о HTML.
- Читайте, задавайте вопросы на Форумах. Здесь вы можете получить совет эксперта и многому научиться.
- Наконец, вам нужно - когда вы почувствуете в этом необходимость - продолжить изучение CSS в нашем Учебнике CSS.
- Остаётся только пожелать вам долгих часов общения с вашим новым знакомым - HTML.
- Смотрите себя в Internet :-)