Лучший учебник html. Основы html

Учебник html - это очень полезная штука, т.к. не зная этот самый язык html, практически невозможно <u>создать свой сайт</u> в сети. С html начинается путь любого вебмастера.

Еще совсем недавно, я сам перерыл весь интернет, в поисках толкового, написанного для начинающих, учебника html. Найти хороший обучающий материал, с большим количеством примеров, в котором бы доходчиво объяснялись основы html, оказалось довольно сложно. Поэтому было принято решение написать свой, собственный учебник html, с примерами и объяснениями, расчитанными для начинающих... В нем, я попытался разложить основы html по полочкам.

Учебник html - Содержание

- <u>Введение в язык html</u> Краткое вступление к учебнику,основные понятия и термины.
- <u>Урок 1: Структура web-документа. Вставка комментария.</u> Здесь мы рассмотрим основы html, а также научимся вставлять комментарии в тело документа.
- <u>Урок 2: Создаем свою первую страничку</u>
 В этом уроке Вы создатите свою первую страничку по всем правилам.
- <u>Урок 3: Работаем с текстом</u>
 В этом уроке изложены базовые средства форматирования текста.
- <u>Урок 4: Продолжаем работать с текстом</u> Здесь мы научимся делать текст жирным, курсивом, подчеркнутым, перечеркнутым, маленьким, большим и.т.д.
- <u>Урок 5: Заканчиваем с текстом</u> В этом уроке мы подробно разберем создание списков в языке html.
- <u>Урок 6: Покоряем html ссылки</u> Тут Вы научитесь создавать все возможные виды ссылок.
- <u>Урок 7: Работаем с изображениями</u>
 В этом уроке вы увидете, как легко, оказывается, вставлять изображения на страницу.
- Урок 8: Цвет фона и текста Зпось Вы узизото, как изменать цвет фон
- Здесь Вы узнаете, как изменять цвет фона и цвет текста в языке html.
- <u>Урок 9: Разберемся с таблицами в html</u> Тут Вы научитесь конструировать html таблицы, это очень важная часть языка html.
- <u>Урок 10: Что такое html формы и зачем они нужны.</u>
 Здесь мы поговорим о том, что такое формы, посмотрим пример готовой формы, разберем основной синтаксис.
- <u>Урок 11: Заканчиваем разбираться с формами</u>
 Заканчиваем то, что начали в предыдущем уроке :)
- <u>Урок 12: Смотрим форму в действии</u>.
 В этом уроке я расскажу Вам как увидеть вашу html форму в действии.
- <u>Урок 13: Пару слов про голову документа.</u>
 Здесь мы разберем такие понятия как название документа, ключевые слова и описание документа.
- <u>Завершение</u> Советы и рекомендации по дальнейшему обучению...

Введение

Рад, что вы решили приступить к изучению моего <u>учебника по html</u>. Поверьте мне, в нем нет ничего сложного, и уже через час вы создадите свою первую страничку. **HTML** – это язык разметки документов в среде WEB. То, что вы видите при просмотре страницы в Internet, это интерпретация вашим браузером HTML-текста. Чтобы браузер правильно отображал форматирование к примеру текста т.е. разделял его на абзацы, выделял цитаты, заголовки, списки и.т.д. ему надо как-то сообщить , что мол это заголовок, а это – параграф и.т.д. Этим как раз и занимается язык html. Чтобы увидеть HTML-коды страницы в Internet, кликните правой кнопкой мыши по странице, в выпавшем меню выберите пункт - view source (или "просмотр HTML кода"). Не пугайтесь! Я еще раз повторюсь – на самом деле, все очень просто!

Что понадобится нам для обучения?

Браузер и Блокнот(или аналогичный простой редактор) - вот всё, что вам необходимо для работы с данным учебником . Раз вы просматриваете эту страничку – значит браузер у вас уже есть , ну а блокнот я думаю найти не составит труда (Пуск – Программы - Стандартные - Блокнот).

Почему простой текстовый редактор идеально подходит для изучения HTML и CSS? да потому, что он не изменяет вводимый вами код. Так вы быстро продвинетесь, а ошибки будут только вашими, а не программными.

Возможно вы уже слышали, или даже использовали такие программы, как Microsoft FrontPage, Adobe Dreamweaver, которые упрощают работу с кодом. Пока что забудьте об этих программах! пока учимся в блокноте!

пока что забудьте об этих программах: пока учимся в олокноте: Па, без сомнений Мы будем ими пользоваться, но только после то

Да, без сомнений Мы будем ими пользоваться, но только после того, как пройдем коротенькие курсы по html и css. Потом просто скачаете у меня на сайте специальный видео-курс по Adobe Dreamweaver и будет вам счастье...

Весь ли здесь HTML ?

Я постарался собрать только ту часть языка, которая действительно используется (из личного опыта создания более 10 сайтов), потому как более чем 40% используется очень редко. Тем более, в дальнейшем мы будем пользоваться Adobe Dreamweaver, и весь труднозапоминающийся код он напишет за нас. Главу по фреймам я выкинул полностью, советую Вам вообще никогда ими не пользоваться, т.к. это уже очень устаревшая технология и рейтинга это Вам не добавит!

Если Вам необходима более полная информация по html, скачивайте книги из соответствующего раздела у меня на сайте.

3 термина используемые по ходу обучения

Тег – оформленная единица HTML-кода. Например, <html >, <body>,<h1>,<h2> и так далее. Все тэги имеют одинаковый формат: они начинаются знаком "<" и заканчиваются знаком ">".Обычно имеются два тэга – открывающий и закрывающий. Различие в том, что в закрывающем имеется слэш "/". Желательно вводить все теги в нижнем регистре(маленькими буквами), хотя не обязательно.

Приведу пример :

<h1></h1>	Крупный заголовок
<h2></h2>	Заголовок чуть поменьше

В браузере это будет выглядеть следующим образом:



Крупный заголовок

Заголовок чуть поменьше

يدعون والمستحل والمحاصرين والمحاصرين والمحاصر والمحاصر فالمحاص والمحاص والمحاص والمحاص والمحاص والمحاص والمحاص

Как вы наверное догадались тег **<h1>** означает заголовок первого уровня, а тег **<h2>** второго уровня. В начале тег открывается, затем идет его содержимое, потом он закрывается. Все довольно просто!

Атрибут –это своего рода дополнительная информация. Атрибуты всегда записываются внутри тега, затем следует знак равенства и детали атрибута, заключённые в двойные кавычки.

Приведу пример:

<h1> Крупный заголовок </h1><h2 align="center"> Заголовок чуть поменьше </h2>

В браузере это будет выглядеть следующим образом:



Крупный заголовок

Заголовок чуть поменьше

had a short when a short the second and a short the second second second and the second s

Как видите, я дописал в теге **<h2>** атрибут align="center", что означает выравнивание по центру, в результате чего содержимое тега выравнялось по центру ячейки. В теге **<h1>** атрибута align (выравнивание) нет, и поэтому выравнивание происходит по умолчанию(по левому краю).

Элемент – понятие, введенное чисто для удобства объяснения. Например, элемент HEAD состоит из двух тегов – открывающего <HEAD> и закрывающего </HEAD>. Таким образом, элемент – более общее понятие, обозначающее пару тегов и участок документа между тегами, на который распространяется их влияние.

Урок 1: Структура web-документа. Вставка комментария.

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, должен начинаться со строки объявления версии HTML <!DOCTYPE...>, которая обычно выглядит примерно так:

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

В данном случае мы говорим браузеру, что HTML соответствует международной спецификации версии 4.01. Не пытайтесь запомнить эту строчку наизусть, главное - вы должны знать, что она необходима. Ведь именно она помогает сделать сайт, который будет одинаково смотреться во всех браузерах(обычно это три основных Internet Explorer,Opera, Firefox).

Далее обозначается начало и конец документа тегами <html> и </html> cоответственно. Внутри этих тегов должны находиться теги головы (<head></head>) и тела документа(<body></body>).

Обычно основой головы документа является элемент TITLE - заголовок документа. Также там содержится вся техническая информация, различные таблицы стилей и.т.п. А основное содержимое: текст, таблицы, картинки, - находится в теле документа.

На этой смешной картинке я попытался схематически изобразить структуру:

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



Как вы видите голова находится над телом, поэтому никогда не размещайте голову документа в теле документа (или наоборот). Сначала закрываем голову документа </HEAD>, и лишь затем открываем тело <BODY>. И еще, у документа одна голова и одно тело, и не стоит пытаться создавать их большее количество.

Вставка комментария: <!-- -->

Очень полезным при создании сайтов является вставка комментария. Почему? да потому, что через месяц после создания, вам будет трудновато разобраться что к чему, ведь кода будет очень много. А если вы будете комментировать свои действия - тогда разобраться будет значительно легче! Т.к. это <u>учебник html с примерами</u>, давайте к ним и обратимся:)

Пример:

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

<html> <head>

<title> Cайт о собаках </title>

</head>

<!-- начнем работать с телом документа -->

<body>

<!--вставляем таблицу с перечнем основных пород собак-->

</body>

<!--закончили с телом документа-->

</html>

Вы уже наверное догадались, что закомментированный текст нужен только для Вас, т.е. при просмотре документа через браузер его видно не будет.

В начале комментарий нужно открыть тегом <!-- затем вписать текст, и закрыть тегом --> .

Примечание: тег <!-- --> внутри элемента TITLE не действует.

Урок 2: Создаем свою первую страничку

Давайте сразу учиться все делать правильно. Хоть это и самая примитивная страничка создадим ее по всем правилам(они пригодятся при работе с Adobe Dreamweaver). **ШАГ 1:**

Создайте на своем компьютере папку с названием вашего будущего сайта, ну к примеру в данном случае актуально будет назвать папку **obuchenie_html**. Внутри папки создайте еще одну папку с названием **www**, а внутри нее еще одну для изображений с названием **img** (позже вставим туда картинки). Для наглядности я нарисовал вам приблизительную структуру, получившуюся в результате:



Пока поверьте мне на слово, дальше когда начнете изучать Dreamweaver все поймете. **ШАГ 2:**

Зайдите в созданную на предыдущем шаге папку **www** и создайте внутри нее текстовый документ. Для этого зайдите в нее, кликните правой кнопкой мыши в выпадающем меню выберите Создать - Текстовый документ. Перед собой вы должны видеть примерно следующее:

img	Вид	•
	Упорядочить значки Обновить	>
Î	Настройка вида папки	
	Вотевить Вотевить арлык Отменить удаление Ctrl+Z	
1	Создать	Папку
1	Свойства	Дрлык
		Портфель Точечный рисунок Документ Microsoft Word Документ Microsoft Office Acces Лриложение Microsoft Office Acces Ловентация Microsoft Office Acces Ловентация Microsoft Office Publishe Документ Microsoft Office Publishe Документ Microsoft Office Publishe Документ Microsoft Office Publishe
		🧧 Текстовый документ

ШАГ 3:

Теперь, следуя предыдущему уроку напишите в Блокноте текст простейшей странички:

Приведу пример:

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

<html>

<head>

<title> Это моя первая страничка! </title>

</head>

<body>

<!--сейчас я напишу свою первую строчку текста!--> Ура!!!! Я создал свою первую страницу на html!!!

</body>

</html>

Готово ! Вы создали свою первую настоящую веб-страницу!

ШАГ 4:

Теперь сохраните ваш документ как "index.htm" (расширение ".htm" указывает, что это HTML-документ. ".html" даст тот же результат. Это не имеет значения). Для этого в меню Файл выберите Сохранить Как:



В пункте Тип файла выберите "Все файлы".

После сохранения можете удалить текстовый документ.

ШАГ 5:

Теперь если вы запустите получившийся файл index.htm увидите примерно следующее:



Для того, чтобы внести изменения в страничку, Вам нужно открыть данный файл через обычный блокнот. Либо же, можно менять прямо в браузере, но только в браузере Opera. Поэтому, если у Вас установлен именно он, то Вы можете просто запустить свою страничку, кликнуть правой кнопкой мыши, в выпадающем меню выбрать "Исходный Код". Там можете менять все, что угодно. А затем, просто нажать на кнопку "Применить изменения".

Как видите действительно в заголовке документа написано "Это моя первая страничка!" а комментарии действительно не отобразились в браузере!

Дело осталось за малым! Научиться форматировать текст, таблицы и.т.п.

Урок 3: Работаем с текстом

В предыдущем уроке мы создали свою первую простейшую страничку. В этом уроке мы подробнее остановимся на свойствах текста, научимся выделять абзацы, заголовки и т.д.

Если вы возьмете и скопируете текст данного урока от начала и до первой картинки, и вставите его в свою первую страничку, то вы увидите примерно следующее:



Ура-а-а-а-а!! Это моя первая страничка!Урок 3: Работаем с текстом В предыдущем уроке мы создали свою первую простейшую страничку. В этом уроке мы подробнее остановимся на свойствах текста, научимся выделять абзацы, заголовки и.т.д. Если вы возъмете и выделите текст данного урока от начала и до первой картинки, и вставите его в свою первую страничку, то вы увидете примерно следующее:

and the second and the second second second and the second s

Как вы заметили текст идет слитно, без разделений на абзацы, также ничего не выделено заголовком, короче форматирование отсутствует! Т.е. чтобы Браузер правильно отображал текст, необходимо задать его форматирование.

Основными элементами форматирования текста в html являются:

- <u>P</u>	Используется для разбивания текста на параграфы
- <u>H1,H2,H6</u>	Применяются для создания заголовков 1,26 уровней
- <u>BR</u>	Используется для переноса строки
- <u>DIV</u> , <u>SPAN</u>	Используются для выделения части документа определенным способом.

P - используется для разметки параграфов в html документах.

Атрибуты:

ALIGN - определяет способ горизонтального выравнивания параграфа.

Возможные значения: left, center, right, justify. Это соответственно по левому краю, по центру, по правому краю, и по ширине .По умолчанию имеет значение left.

Приведу пример:

Если внутри тела документа написать следующее:

То при просмотре в браузере получим следующее:



Этот параграф по центру

Этот по левому краю

Этот тоже по левому (т.к. по умолчанию)

Этот по правому краю

В этом параграфе текст будет выравниваться по ширине (одновременно по левому и правому краям документа). Не понимающие justify браузеры будут выравнивать текст по левому краю

anna ann N. A.

Примечание: следите чтобы содержимое в кавычках было написано без пробелов, т.е. а не иначе не будет работать!

Н1,H2,...H6 - Применяются для разбивания текста на смысловые уровни - разделы и подразделы. Существует шесть уровней заголовков, различающихся по величине шрифта.

Атрибуты:

ALIGN - определяет способ горизонтального выравнивания заголовков. Возможные значения: left, center, right .По умолчанию - left.

Приведу пример:

Если внутри тела документа написать следующее:



То при просмотре в браузере увидим:



Большой заголовок

Заголовок поменьше

Еще меньше

Совсем маленький

Малюсенький заголовочек



Примечание: закрывающий тег обязателен!

BR - данный элемент осуществляет перенос строки. Не имеет закрывающего тега.

Приведу пример:

Если внутри тела документа написать следующее:

Начинается первая строка **
** теперь идет вторая **
** а вот уже и третья А это параграф и внутри него тоже можно использовать тег **
** переноса строки.Его надо использовать всегда, когда надо перенести строку:)

Увидим следующее:



Как вы наверное заметили, при переносе строки не начинается новый параграф! Перенос- он и есть перенос!

DIV - В современном сайтостроении используется как удобный контейнер для блоков html кода страницы, которым легко динамически манипулировать – перемещать, регулировать отступы, скрывать и т.п. Обязателен закрывающий тег!

Атрибуты:

ALIGN - определяет способ горизонтального выравнивания контейнера. Возможные значения: **left, center, right, justify**. Это соответственно по левому краю, по центру, по правому краю, и по ширине .По умолчанию имеет значение **left**.

Допустим нам нужно выровнять первые две строчки текста из предыдущего примера по правому краю, не выделяя при этом их в абзац. Как это сделать?

<div align="right">
Haчинается первая строка
 теперь идет вторая
</div>
</div>
a вот уже и третья

A это параграф и внутри него тоже можно использовать тег
 переноса строки.Его
надо использовать всегда, когда надо перенести строку:)

Что мы увидим:



А это параграф и внтури него тоже можно использовать тег переноса строки. Его надо использовать всегда, когда надо перенести строку.)

and a second descent and a

Т.е. мы взяли две строчки кода, заключили их в контейнер **DIV** и выравняли его по правому краю! Так же само, можно заключить в контейнер и 20 и 50 и более строчек и манипулировать ими.

Примечание: Находящиеся между начальным и конечным тегами текст или HTMLэлементы выделяются как бы в отдельный параграф(но отступы гораздо меньше).

Например:

Пока что вы читаете текст на первой строке <div> Но теперь уже на второй </div> А теперь уже на третей

Результат:



Как видите текст содержащийся в элементе DIV выделился в отдельную строку!

SPAN - Используется для выделения части информации и придания ей различных стилей. Закрывающий тег обязателен! Сам по себе, без использования стилей, элемент SPAN никакого смысла не имеет!

Например:

Хотите пользоваться **** Adobe Dreamweaver **** - тогда изучите хотя бы основы html и **
** тогда Вы без проблем разберетесь с этой замечательной программой!

Что мы увидим:

<u>Ф</u> айл	Правка	Вна	Журнал	<u>З</u> акладки	<u>И</u> нструменты <u>С</u> правка		
<	- 🦚	C	⊗ 🏠	file:/	//C:/Documents%20a En	G• Google	Q

Хотите пользоваться Adobe Dreamweaver - тогда изучите хотя бы основы html и тогда Вы без проблем разберетесь с этой замечательной программой!

when any a part when a week a second a second and the second have a second to a second have a second

Как видите, как будто мы ничего и не выделяли! Но если прописать какой нибудь стиль этому тегу, то он начнет работать!

Хотите пользоваться <**span style="color:red;"** > Adobe Dreamweaver </**span>** - тогда изучите хотя бы основы html и <**br**>

тогда Вы без проблем разберетесь с этой замечательной программой!

В данном случае я прописал стиль красного цвета, подробнее о стилях в курсе CSS.



Таким образом с помощью элемента **SPAN** можно хоть каждой букве в тексте присвоить разный стиль(цвет, размер, и.т.д.)

Урок 4: Продолжаем работать с текстом

Чтобы придать тексту ту или иную гарнитуру в html используются такие элементы:

- <u>STRONG</u>	Используется для выделения текста жирным
- <u>EM</u>	Используется для выделения текста курсивом
- <u>U</u>	Выделение текста подчеркиванием
- <u>s</u>	Перечеркивание текста
- <u>SUP</u>	Создание верхнего индекса
- <u>SUB</u>	Создание нижнего индекса
<u>- FONT</u>	Изменение цвета, типа, размера шрифта
- <u>HR</u>	Вставляет в текст горизонтальную разделительную линию

STRONG - Выделяет текст, заключенный между открывающим и закрывающим тегами, жирным шрифтом. Раньше везде использовался тег **SOLD** (или **SP**), в принципе его можно использовать и сейчас, но это не приветствуется поисковыми системами.

Если в теле документа написать:

 Этот текст будет выделен жирным
И этот тоже, но этим тегом лучше не пользоваться!

В браузере увидим:



Как видите, оба этих тега действуют одинаково. В программе Adobe Dreamweaver на панели форматирования текста написано "**B**"



(просто это название более старое и большинство веб-мастеров привыкли к нему), хотя при нажатии на эту кнопочку в коде пишется тег ****.

EM (Emphasis) - Выделяет текст, заключенный между открывающим и закрывающим тегами, курсивом. Также аналогичен по действию тег <I>, однако его лучше не использовать т.к. это не приветствуется поисковыми системами. В программе Adobe Dreamweaver на панели форматирования текста написано "I" хотя в коде пишется EM.

Приведу пример:

 Этот текст будет выделен курсивом
 <i> И этот тоже, но лучше пользоваться тегом EM</i>

В браузере увидим:



^{[▶] U(Underline)</mark> - Отображает помещенный между открывающим и закрывающим тегами текст как подчеркнутый:}

<u> Этот текст будет подчеркнутым!!! </u>

Увидим в браузере:



[▶] S(Strike) - Текст, помещенный между открывающим <s> и закрывающим </s> тегами, будет перечеркиваться!

Например:

Внимание акция! новая цена **<s>** 50 000 **</s>** 20 000 рублей!

Результат:



SUP (Superscript) - Отображает текст, заключенный между открывающим ^{и закрывающим} тегами, как верхний индекс от основного текста.

Например:

2 ² = 4 ;	
2 ³ = 8 ;	
2 4 = 16 ;	

И что получим:



SUB(Subscript) - Отображает текст, заключенный между открывающим _{и закрывающим} тегами, как нижний индекс от основного текста.

Приведу пример:

Формула спирта в химии - C ₂ H ₅ OH

И что мы получим:



► FONT - Позволяет изменять цвет, размер и тип шрифта текста, находящегося между открывающим и закрывающим тегами.

Атрибуты:

SIZE - Определяет размер шрифта. Возможные значение - 1, 2, 3, 4, 5, 6, 7.

COLOR - Определяет цвет текста. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

FACE - определяет используемый шрифт. Используйте Times New Roman, Arial, Tahoma, Courier, Courier New. Они установлены почти у всех. Иначе гарантий нет!

Если в теле документа написать:

```
Это обычный текст <br><FONT SIZE="+2" COLOR="red">Увеличенный красный шрифт</FONT><br><br><br><br><br><fONT SIZE="3" FACE="Courier New" COLOR="Violet">Моноширинный фиолетовый текст 3 размера</FONT>
```

При просмотре в браузере увидим:



Если написано **SIZE="+2"** то это означает увеличить на 2 единицы шрифт по сравнению со стандартным. Обычно стандартный размер шрифта равен **3**.

Примечание: В дальнейшем(после изучения css) лучше вообще все что связано с типом, размером и цветом шрифта делать через таблицы стилей. Это гораздо удобнее, тем более многие элементы форматирования текста расположенные внутри элемента FONT, работают некорректно.

^и HR- служит для вставки в текст горизонтальной линии. Закрывающего тега нет!

Атрибуты:

WIDTH – определяет длину линии в пикселах или процентах от ширины окна браузера.

SIZE – толщина линии в пикселах.

ALIGN – определяет выравнивание горизонтальной линии. Атрибут может принимать следующие значения:

left – выравнивание по левому краю документа.

right – выравнивание по правому краю документа.

center – выравнивание по центру документа (используется по умолчанию).

NOSHADE – определяет способ закраски линии как сплошной. Атрибут является флагом и не требует указания значения. Без данного атрибута линия отображается объемной.

COLOR – задает цвет линии. Можно использовать либо RGB-значение в шестнадцатиричной системе, либо один из 16 базовых цветов. Атрибут работает только в Internet Explorer.

Например:

Текст до линии <hr noshade width="50%" align="left"> После линии
 А вот пример линии толщиной 2px и без флага noshade <hr width="50%" align="left" size ="2">

Что мы увидим:



Урок 5: Заканчиваем с текстом

В этом уроке мы разберем такую немаловажную часть языка html как списки. Различают списки двух видов: упорядоченные (пронумерованные) и неупорядоченные (непронумерованные). Единственным отличием является то, что перед пунктами неупорядоченных списков обычно ставятся символы-буллеты (bullets), например, точки, ромбики и т.п., в то время как пунктам упорядоченных списков предшествуют их номера.

Элементы:

- <u>UL</u>	Создает неупорядоченный список
- <u>OL</u>	Создает упорядоченный список
- <u>u</u>	Создает пункты списка внутри элементов OL или UL

²¹ UL (Unsorted List) - Создает неупорядоченный список. Обязательно наличие закрывающиего тега, причем между начальным и конечным тегами должны присутствовать один или несколько элементов <u>LI</u>, обозначающих отдельные пункты списка.

Например:

Первый пункт списка
Второй пункт списка
Третий пункт списка
Четвертый пункт списка

Что мы увидим в браузере:



² OL (Ordered List)- Создает упорядоченный список. Между начальным и конечным тегами должны присутствовать один или несколько элементов <u>LI</u>, обозначающих отдельные пункты списка.

Атрибуты:

START – определяет первое число, с которого начинается нумерация пунктов. (только целые числа). Если не указывать, начинается с начала.

ТҮРЕ – определяет стиль нумерации пунктов списка.

Возможные значения:

"А" – заглавные буквы А, В, С ... "а" – строчные буквы а, b, с ... "I" – большие римские числа I, II, III ...

"i" – маленькие римские числа i, ii, iii ...

"1" – арабские числа 1, 2, 3 ...

Значение по умолчанию <OL TYPE="1">.

Приведу пример:



наличие хорошей посещаемости
наличие цифрового продукта для продажи
наличие блоков рекламы
пример списка не с начала
Чтобы стать успешным в интернете:
<ol start="2" type="1">
писать информацию полезную для людей
постоянно пополнять сайт новым содержимым
первый пункт придумай сам
0

В результате получим:

🕲 Моя	первая	страни	iųa I - Moz	illa Firefox	efox			
<u>Ф</u> айл	Правка	Вид	Журнал	<u>З</u> акладки	Инструменты Справка	e C		
<	\$ ·	C	3 🟠	📄 file:/.	VC:/Documents%20a 📴	• 🔊	G - Google	Q
🥐 Hav	альная ст	раница	🔯 После	дние заголо	вки			

Чтобы создать сайт на домашнем компьютере необходимо:

- 1. Выучить html
- 2. Выучить css
- 3. Ознакомиться с php

Чтобы создавать сайты быстро желательно:

- I. знания программы Adobe Dreamweaver
- II. знания программы TOpStylePro
- Ш. Знания Photoshop

Чтобы зарабатывать на сайте желательно:

- а. наличие хорошей посещаемости
- в. наличие цифрового продукта для продажи
- с. наличие блоков рекламы

Чтобы стать успешным в интернете:

- 2. писать информацию полезную для людей
- 3. постоянно пополнять сайт новым содержимым
- 4. первый пункт придумай сам

The second second and the second s

^Ш LI (List Item) - Создает пункт в списке. Располагается внутри элементов <u>OL</u> или <u>UL.</u> Закрывающий тег писать желательно, но не обязательно, это уже решать Вам.

Атрибуты:

VALUE – изменяет порядок нумерации элементов списка. Используется только если элемент **LI** находится внутри элемента **OL**. В качестве значения указывается порядковый номер элемента.

Приведу пример:

Чтобы создать сайт на домашнем компьютере необходимо: <!--так как тип не указали будет использоваться по умолчанию--> Bыучить html <!--сейчас нумерация пойдет с пятого номера--> Выучить css Oзнакомиться с php

Результат в браузере:

TONI	Правка	Вид	Журнал	Закладки	Инструменты	<u>С</u> правка		Q				
ج ج	ا ≪	e	8 6	file:/	///C:/Documents3	20a En 🔹 🕨	G . Google	Q				
🥐 Ha	чальная ст	раница	🔂 Посля	адние загол	овки							
Чтобы создать сайт на домашнем компьютере необходимо:												
Чтоба	ы создат	ъ сай	г на дом	ашнем ко	мпьютере не	еобходимо:						
Чтобі 1.	ы создат Выучит	ъcай ъhtml	г на дом	ашнем ко	мпьютере н	еобходимо:						
Чтобі 1. 5.	ы создат Выучит Выучит	ъ сай ъ html ъ css	г на дом	ашнем ко	мпьютере н	собходимо:						

Урок 6: Покоряем html ссылки

В первую очередь давайте поговорим о ссылках - переходах с одной страницы на другую. Ссылки это базовый элемент без которого невозможно представить интернет.

Создать html ссылку очень просто. Для этого используется простенький элемент с одним атрибутом. Ну вот например:

Это ссылка на сайт zvirec.com

будет выглядеть в браузере:

Это ссылка на сайт zvirec.com

Элемент **a** (**ancor**) является как бы якорем, т.е. текст, заключенный между открывающим <**a**> и закрывающим <**/a>** тегом , будет текстом ссылки.

Атрибут href это сокращение от "hypertext reference/гипертекстовая ссылка", специфирует место, на которое выполняется переход по данной ссылке - обычно это internet-адрес и/или имя файла.

Если нужно сделать ссылку между собственными страницами?

Сделать html ссылку между страницами одного и того же веб-сайта гораздо проще. Например, если у нас есть две страницы(к примеру **page1.htm** и **page2.htm**) расположенные в одной папке, то код ссылки с page1 на page2 будет выглядеть так:

 Для перехода на page2 щелкни здесь!

Т.е. надо просто написать название страницы, на которую мы хотим перейти.

Если страница page 2 находится в подпапке "subfolder", код ссылки выглядит так:

Для перехода на page2 щелкни здесь!

Так будет выглядеть код html ссылки со страницы page 2 на page1(в обратную сторону):

 Для перехода на раде1 щелкни здесь!

Т.е. сочетание "../" указывает как бы на папку, расположенную на один уровень выше от данной позиции файла, с которого делается ссылка.

А если надо сделать ссылку внутри страницы?

Бывают случаи когда необходимо сделать ссылку с начала страницы в конец или наоборот, или к примеру с оглавления на главы и.т.д. Сделать ее очень просто! Необходимо пометить место документа следующей конструкцией:

<!-- этот способ используется при маркировании заголовков --> <h2 id="razdel1">Раздел 1</h2> <!-- а такую метку можно поставить везде где понадобится--> <h2> Раздел1: </h2>

Теперь вы можете ссылаться на помеченную область(в данном случае Раздел1) простым указанием ее имени после значка #.

Вот как будет выглядеть ссылка на раздел1:

 Ссылка на Раздел 1

Приведу более наглядный пример:

<h1 > Оглавление </h1>
Раздел 1: как стать богатым
Раздел 2: как стать счастливым
Раздел 2: как стать счастливым<h2 id="razdel3">Раздел 3: как быть здоровым<h2 id="razdel1">Раздел 1: как стать богатым </h2>Для того, чтобы стать богатым необходимо очень много трудиться <h2 id="razdel2">Раздел 2: как стать счастливым </h2>Для того чтобы стать счастливым, нужно использовать каждую минуту...<h2>Раздел3:Для того чтобы быть здоровым нужно много заниматься физкультурой...

Вот результат в браузере:

Оглавление Раздел 1: как стать богатым Раздел 2: как стать счастливым Раздел 3: как быть здоровым Раздел 1: как стать богатым

Для того чтобы стать богатым необходимо очень много трудится

Раздел 2: как стать счастливым

Для того чтобы стать счастливым, нужно использовать каждую минуту...

Раздел3: как быть здоровым

Для того чтобы быть здоровым нужно много заниматься физкультурой...

А можно сделать ссылку на почту?

Конечно можно. Вот пример ссылки на e-mail сайта zvirec.com .

Написать письмо админу zvirec.com

В браузере будет выглядеть:

Написать письмо админу zvirec.com

Как вы видите, используется mailto потом двоеточие и емаил получателя. При нажатии на ссылку запустится почтовая программа-клиент с заполненным полем имени получателя.

Парочка дополнительных атрибутов:

TARGET - указывает где открывать страницу на которую ведет html ссылка. По умолчанию она открывается в том же окне. Если хочется чтобы открывалась в новом, следует написать **target ="_blank"**.

TITLE - указывает заголовок ссылки, который появляется при наведении на нее.

Смотрите пример:

Это ссылка на сайт zvirec.com, откроется в новом окне

Эта - тоже на zvirec.com. При наведении появится заголовок.

Смотрите на результат:

Это ссылка на сайт zvirec.com, откроется в новом окне

Эта - тоже на zvirec.com. При наведении появится заголовок.

Ну вот в принципе и всё. Можете сильно не зацикливаться, в программе Adobe Dreamweaver любая ссылка делается одним нажатием мышки. Главное для Вас, понять что к чему, а зазубривать не обязательно...

А можно ли изменить цвет ссылок?

Конечно можно. Чтобы изменять цвет ссылок во всем документе, существуют специальные атрибуты элемента BODY.

LINК - цвет просто ссылок.

ALINK(Active Link) - цвет активных ссылок (активная означает в момент нажатия на нее) VLINK(Visited Link) - цвет уже посещенных ссылок

Все цвета задаются либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

К примеру если при открытии тела документа, элементу body прописать:

<body link="red" vlink="green" alink="white">

то все ссылки в данном документе станут красными, уже посещенные ссылки станут зелеными а ссылки в момент нажатия будут белыми.

А если нужно чтобы в каком -то месте ссылка имела другой цвет? например по всему документу красные, а именно в одном месте зеленая?

Тогда нужно внутри html ссылки прописать уже знакомый элемент font с атрибутом цвета:

Черная ссылка

Если сделать так, то данная ссылка будет черной.

Урок 7: Работаем с изображениями

Изображения - это неотъемлемая часть любого сайта в сети интернет. Они используются везде, поэтому давайте разберемся что к чему. 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)

Пара слов о форматах:

GIF - использует всего 256 цветов и соответственно лучше подходит для рисунков с малым кол-вом оттенков. Этот формат поддерживает прозрачность изображений. JPEG - формат изображений, который использует до миллиона цветов. Обычно используется для фотографий и качественной графики(с огромным количеством оттенков).

PNG - сравнительно новый формат . По многим параметрам превосходит JPEG и GIF: миллионы цветов и эффективное сжатие. Также поддерживает прозрачность.

В каком формате брать изображения - дело Ваше, однако старайтесь добиться максимального качества при минимальном размере.

Вставить изображение на страницу очень просто. Вот пример если оно лежит в той же папке что и страница.

В результате мы увидим:



Что нам понадобилось: элемент IMG не имеющий закрывающего тега и атрибут SRC (сокращение от source - положение) который указывает где находится изображение. Вы легко можете вставлять изображения, размещенные в других папках или на других сайтах. Просто укажите правильный путь (примерно так как при созданиии ссылок).

Вот еще несколько примеров с комментариями:

<!-- если бы изображение находилось в папке images --> <!-- если б находилось на сайте www.zvirec.com --> <!-- если б находилось на сайте www.zvirec.com в папке images -->

Вот еще очень необходимые атрибуты:

ALIGN - определяет способ выравнивания картинки по горизонтали. Очень полезное свойство при обтекании картинки текстом. Обычно используют LEFT (выравнивание по левому краю, текст будет обтекать справа) и **RIGHT** (выравнивание по правому краю, текст обтекает слева). Если на странице есть текст, то это обязательное свойство.

HSPACE и VSPACE - отступы в пикселях по горизонтали и по вертикали от картинки до других объектов документа. Легко запомнить название если взять и просто перевести с английского. HSPACE - Horizontal Space - горизонтальный отступ и VSPACE - Vertical Space - вертикальный отступ.

НЕІGHT и WIDTH - высота и ширина изображения в пикселах. Золотое правило webмастера – всегда явно задавать размеры картинки в атрибутах HEIGHT и WIDTH, резервируя тем самым место в окне браузера еще до загрузки изображения. В противном случае документ при загрузке каждой картинки будет заново перерисовываться. А на медленных машинах (или при подключении к сети через модем) это смотрится просто отвратительно. Но в принципе, можно и не задавать размеры, просто будет дольше загружаться. И еще, не советую Вам искажать реальные размеры картинки.

Ладно, лучше все смотреть на примерах:

<!-- первый пример с отступами и выравниванием по левому краю-->

Любой Интернет бизнес немыслим без сайта. Так что первым делом не мешало бы научиться создавать сайты. В данном случае фраза "научиться создавать сайты" будет означать - знать на хорошем уровне язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS, которые значительно расширяют возможности HTML и позволяют создавать действительно стильные и качественные сайты.

<!-- второй пример с отступами и выравниванием по правому краю--> Любой Интернет бизнес немыслим без сайта. Так что первым делом не мешало бы научиться создавать сайты. В данном случае фраза "научиться создавать сайты" будет означать - знать на хорошем уровне язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS, которые значительно расширяют возможности HTML и позволяют создавать действительно стильные и качественные сайты.

<!--третий пример без отступов, с выравниванием по левому краю--> Любой Интернет бизнес немыслим без сайта. Так что первым делом не мешало бы научиться создавать сайты. В данном случае фраза "научиться создавать сайты" будет означать - знать на хорошем уровне язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS, которые значительно расширяют возможности HTML и позволяют создавать действительно стильные и качественные сайты.

Вот результат браузере:

		<u>С</u> правка	нструменты	Закладки	Курнал	Вид	Правка	<u>Ф</u> айл
Q	G . Google	20a En 🔹 🕞	/Documents%2	file://	3 🔂	C	⇒ ·	<
	G . Google	20a Eng 🔹 🔛	:/Documents%2	file://		C	- (m)	 • •



Любой Интернет бизнес немыслим без сайта. Так что первым делом не мешало бы научиться создавать сайты. В данном случае фраза "научиться создавать сайты" будет означать - знать на хорошем уровне язык гипертекстовой разметки НТМL и каскадные таблицы стилей CSS, которые значительно расширяют

возможности HTML и позволяют создавать действительно стильные и качественные сайты.

Любой Интернет бизнес немыслим без сайта. Так что первым делом не мешало бы научиться создавать сайты. В данном случае фраза "научиться создавать сайты" будет означать - знать на хорошем уровне язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS, которые значительно расширяют возможности HTML и позволяют создавать действительно стильные и



качественные сайты. 💐 Любой Интернет бизнес немыслим без сайта. Так что первым делом не мешало бы научиться создавать сайты. В данном случае фраза "научиться создавать сайты" будет означать - знать на хорошем уровне

язык гипертекстовой разметки HTML и каскадные таблицы стилей CSS, которые значительно расширяют возможности HTML и позволяют создавать действительно стильные и качественные сайты.

a stream the second second

А есть ещё атрибуты, которые нам необходимы?

Вот еще пара полезных атрибутов - ALT и TITLE.

ALT - определяет текст, отображаемый браузером на месте изображения, если браузер не может найти файл с изображением или включен в текстовый режим. В качестве значения задается текст с описанием изображения. Смотрите пример:

В результате увидим:

```
пчела мая!!!
```

Как видите, я поменял название файла, и браузер не смог его найти, поэтому показал текст, который указан в атрибуте ALT.

TITLE - Определяет заголовок картинки, т.е. при наведении на картинку высветится надпись с текстом из атрибута title . Ну вот например:

Попробуйте навести указатель на изображение:



А как можно изображение сделать ссылкой?

Для этого просто вместо текста ссылки, вставляете изображение. Вот например:

```
<a href="http://www.zvirec.com">
<img src="pchela.jpg" width="65" height="59" title="Пчела мая!!! "
border="0" >
</a>
```

Вот результат:



Что еще за **border="0"** спросите Вы, а дело в том, что когда делаешь изображение ссылкой, вокруг него появляется некрасивая рамка(border) и чтобы ее убрать пишут дополнительный атрибут border.

Ну вот впринципе и все с изображениями, как видите ничего сложного нет.

Урок 8: Цвет фона и текста

Мы уже знаем как менять цвет текста, но для этого нам нужно было заключать его в теги **font**, а это не всегда удобно. Иногда, лучше задать цвет текста для всего документа. Также, можно задать и фоновое изображение.

Вот необходимые атрибуты:

BACKGROUND – определяет изображение для "заливки" фона. Значение задается в виде полного URL или имени файла с картинкой в формате GIF или JPG.

BGCOLOR – определяет цвет фона документа.

ТЕХТ – определяет цвет текста в документе.

Все они прописываются для элемента BODY. Значения цветов задаются либо RGBзначением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

Пример1:

задаем фоновый цвет и цвет текста
<body bgcolor="#FFF8D2" text="red"></body>
>Этот текст будет красный, потому что мы изменили цвет текста в теге БОДИ и теперь
весь текст на странице по умолчанию будет красный

В этом абзаце текст будет зеленый, потому что мы заключили его в теги font и
придали соответсвующий цвет
Теперь текст снова будет красный

Результат в браузере:



<body background="fon.jpg" text="red">Этот текст будет красный, потому что мы изменили цвет текста в теге БОДИ и теперь весь текст на странице по умолчанию будет красный Teперь тут тоже красное и только эти слова зеленые Tyт как вы поняли текст тоже красный

Результат в браузере:



Ничего сложного нет. Экспериментируйте, практикуйтесь и все получится!

Урок 9: Разберемся с таблицами в html

Html таблицы - полезная штука . Обычно их используют не только для отображения таблиц как таковых, но и для создания невидимого каркаса страницы, помогающего расположить текст и изображения должным образом. Раньше все сайты имели табличную структуру, сейчас все большую популярность приобретает структура на дивах (с помощью <div> и CSS). Вот классический пример табличной структуры:

	Шапка сайта(логотип и все такое)	
Ссылка 1 Ссылка 2 Ссылка 3 Ссылка 4 Ссылка 5	Основное содержание	Реклама и все такое
Блок копирайта		

Давайте разбираться как построить простейшую таблицу. Для этого нам понадобятся как минимум три элемента.

TABLE - Элемент для создания *html таблицы*. Обязательно должен иметь начальный и конечный теги. По умолчанию html таблица печатается без рамки, а разметка осуществляется автоматически в зависимости от объема содержащейся в ней информации. Сразу скажу, что здесь есть **атрибут BORDER** который задает толщину рамки таблицы в пикселах.

TR (Table Row) - Создает новый ряд таблицы. Закрывающий тег обязателен.

TD (Table Data) - Начинает и заканчивает каждую ячейку ряда html таблицы. Обязателен закрывающий тег.

Давайте попробуем создать таблицу из двух рядов и двух столбцов:

Вот как это будет выглядеть в браузере:

ряд 1 ячейка1 ряд1 ячейка2 ряд 2 ячейка 1ряд 2 ячейка 2

Разобрались? Т.е. таблица начинается с , затем идёт , указывающий на начало нового ряда. В ряду вставлены две ячейки: ряд 1 ячейка1 и лица закрывается и сразу начинается новый ряд . В новом ряду также две ячейки. Таблица закрывается . Все довольно логично

А как можно объединить ячейки?

Для этого существуют данные атрибуты.

COLSPAN – определяет количество столбцов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

ROWSPAN – определяет количество рядов, на которые простирается данная ячейка. По умолчанию имеет значение 1.

Пример 1:

```
ряд 1 ячейка 1+2
>ряд 2 ячейка 1 >ряд 2 ячейка 2
```

Получим в браузере:

ряд 1 ячейка 1+2 ряд 2 ячейка 1ряд 2 ячейка 2

Пример 2:

```
Ячейка 1, ряд 1+2
pяд1 ячейка2
pяд 2 ячейка 2
```

Получим в браузере:

Ячейка 1, ряд 1+2<mark>ряд1 ячейка2</mark> ряд 2 ячейка 2

Ничего сложного нет. Присмотритесь внимательно к трем примерам и все станет ясно.

Вот еще полезные атрибуты:

CELLPADDING – определяет расстояние (в пикселах) между рамкой каждой ячейки html таблицы и содержащемся в ней материалом.

CELLSPACING – определяет расстояние (в пикселах) между границами соседних ячеек таблицы html.

WIDTH – определяет ширину таблицы html. Ширина задается либо в пикселах, либо в процентном отношении к ширине окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в таблице материала.

HEIGHT – определяет высоту таблицы. Высота задается либо в пикселах, либо в процентном отношении к высоте окна браузера. По умолчанию этот атрибут определяется автоматически в зависимости от объема содержащегося в html таблице материала.

Пример 1:

```
pяд 1 ячейка1
pяд 1 ячейка2
pяд 2 ячейка 1
pяд 2 ячейка 2
```

Результат в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Пример 2:

```
pяд 1 ячейка1
/td>pяд 1 ячейка2
/td>
```

Результат в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Пример 3:

ряд 1 ячейка1ряд1 ячейка2
pяд 2 ячейка 1 ряд 2 ячейка 2

Результат в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

И наконец последние четыре атрибута:

ALIGN – определяет способ горизонтального выравнивания html таблицы или содержимого ячеек. Возможные значения: left, center, right. Значение по умолчанию – left. VALIGN – должен определять способ вертикального выравнивания таблицы или содержимого ячеек таблицы .Возможные значения: top, bottom, middle.(прижать к верху, прижать к низу,и установить посередине).

BGCOLOR – определяет цвет фона ячеек таблицы. Задается либо RGB-значением в шестнадцатиричной системе, либо одним из 16 базовых цветов.

BACKGROUND – позволяет заполнить фон таблицы рисунком. В качестве значения необходимо указать URL рисунка.

Примечание: можно хоть каждой ячейке таблицы задать свой цвет фона или фоновый рисунок. **Но:** если вы задаете фоновый цвет (или рисунок) в базовом элементе TABLE то во всех ячейках будет этот фон(рисунок), и если вы захотите в какой-то ячейке поменять, пропишите соответсвующий атрибут именно для этой ячейки.

Приведу пример:

```
<!-- задаем ширину, высоту, рамку, выравнивание по центру и фоновый цвет всей
таблицы-->
<!-- эту ячейку оставляем по умолчанию-->
>ряд 1 ячейка1
```

```
<!-- содержимое горизонтально выравниваем по центру, вертикально - прижимаем
к верху-->
ряд1 ячейка2
<t-- содержимое горизонтально выравниваем по центру, вертикально - прижимаем
к низу-->
ряд 2 ячейка 1
<t-- содержимое горизонтально выравниваем по правому краю, вертикально -
посередине, и меняем фоновый цвет-->
ряд 2 ячейка 2
```

Результат в браузере:

ряд 1 ячейка1	ряд1 ячейка2
ряд 2 ячейка 1	ряд 2 ячейка 2

Еще один пример:

```
<!-- задаем ширину, высоту, рамку, фоновый цвет всей таблицы, выравнивание
оставляем по умолчанию(по левому краю)-->
<!-- горизонтальное-по центру, вертикально по умолчанию(по центру)-->
 ряд 1 ячейка1 
<!-- горизонтальное-по центру, вертикально по умолчанию (по центру) и делаем
фоновый рисунок-->
 ряд1 ячейка2 
<!-- горизонтальное-по центру, вертикально по умолчанию(по центру)-->
 ряд 2 ячейка 1 
<!-- горизонтальное-по центру, вертикально по умолчанию (по центру) и меняем
фоновый цвет-->
ряд 2 ячейка 2
```

Результат в браузере:

ряд 1 ячейка1	Оряд 1 я Сага Саба
ряд 2 ячейка 1	ряд 2 ячейка 2

Если картинка меньше чем ячейка, то она будет дублироваться, как показано в примере. Если больше - то будет отображаться та часть которая влезет :).

НУ вот впринципе и всё по таблицам. Если не получилось запомнить не расстраивайтесь. С практикой все запомнится (как говорят главное знать, где подсмотреть!). В программе Adobe Dreamweaver все делается очень просто.

Урок 10: Что такое html формы и зачем они нужны.

Html формы - это наверное самая сложная часть языка html. Так что наберитесь терпения и давайте потихоньку начинать.

Что такое html формы? Вот смотрите, ниже приведена html форма заказа обучающего диска. По окончании обучения, вы тоже сможете сделать такую.

Введите ФИО:
Введите пароль:
Какой диск вы хотите получить?
CD CD DVD
Какие обучающие курсы вы хотите видеть на диске?
 Курсы по Фотошопу Курсы по Adobe Dreamweaver Курсы по РНР
Выберите способ доставки:
Срочная
Введите адрес для доставки:
Сделать заказ Отмена

Здесь все работает, кроме отправки данных, можете поклацать :)

Ну как видели такое? вот-вот, сейчас везде такое, так что давайте разберемся:

html форма - это всего лишь каркас, созданный при помощи языка html, т.е. мы можем указать браузеру где у нас будет какое поле и что написано на той или иной кнопке. Но для того, чтобы при нажатии на кнопку ваш комментарий добавился в гостевую книгу или данные заказа полетели в офис - нужна иная технология - программа, скрипт, который привязывается к форме. Обычно такие программы делаются на php . Но не расстраивайтесь раньше времени, я дам Вам простейший обработчик для тренировки, а более сложные можно найти в интернете или самому написать.

Давайте рассмотрим все элементы формы из примера выше по-порядку:

Любые элементы формы находятся как бы в **теле формы**. Т.е. у любой формы есть каркас, а уже внутри него вставляются различные элементы формы, и пишется html код.

Пример 1:

<form> <!-- сюда вставляют различные элементы --> </form>

Как видите здесь два базовых тега : открывающий <form> и обязательный закрывающий тег </form> .

Но в этом выше, как бы голый каркас, т.е. без атрибутов. А теперь рассмотрим более реальный каркас *html формы*:

Пример 2:

<form name="forma zakaza" method="post" action="obrabotchik.php"> <!-- сюда вставляют различные элементы --> </form>

Атрибуты:

NAME – определяет имя формы, уникальное для данного документа. Используется только, если в документе присутствует несколько форм.

ACTION – обязательный атрибут. Указывает путь к скрипту(или программе) сервера, обслуживающему данную форму.

METHOD – определяет способ отправки содержимого html формы. Возможные значения GET (по умолчанию) и POST.

Пару слов о методах передачи:

Метод GET используется для передачи различных переменных, или очень коротких сообщений. Информация передается в явном виде через строку браузера, т.е. ее можно перехватить. Например если вы видите в строке набора браузера нечто вроде http://adress.com/lessons.php?rub=28 это значит что передается значение переменной rub равное 28. В html формах обычно не используется.

Метод POST создан специально для передачи текстовых сообщений. Почти всегда применяется в формах. Передает информацию в скрытом виде.

Пока все понятно? тогда давайте начнем разбирать элементы формы:

<form name="primer1" method="post" action="obrabotchik.php"> Введите ФИО:
<input type="text" name="fio" size="30">
 В результате получим:

Введите ФИО:		
Введите пароль:		

Что мы видим в исходном коде? а мы видим следующее:

Элемент INPUT - создает поле html формы (кнопку, поле ввода, чекбокс и т.п.), Элемент не имеет конечного тега.

Основные атрибуты :

ТҮРЕ - определяет тип поля для ввода данных. По умолчанию – это "text". В данном примере еще используется тип "password" который указывает на то, чтоб информация показывалась звездочками.

NAME - определяет имя, используемое при передаче содержания данной html формы на сервер. Если Вы собираетесь куда-нибудь передавать информацию из формы, то обязательно вводите имя.

SIZE - определяет размер поля в символах. По умолчанию имеет значение равное 24. Т.е. если этот атрибут не писать то длина будет равна 24 символа.

Есть еще и такой полезный атрибут как **MAXLENGTH**, который определяет максимальное количество символов, которые можно ввести в текстовом поле. Оно может быть больше, чем количество символов, указанных в атрибуте SIZE. По умолчанию количество символов не ограничено.

Ну вот например:

<form name="primer2" method="post" action="obrabotchik.php"> Введите пароль(максимум шесть символов):
<input type="password" name ="pass" maxlength="6"> </form>

Результат:

Введите пароль(максимум шесть символов):

Ну как работает? А вы попробуйте ввести больше шести символов :)

Еще есть атрибут **VALUE** который определеят что будет по умолчанию написано в поле для ввода.

Смотрите пример:

```
<form name="primer3" method="post" action="obrabotchik.php">
Введите свой e-mail: <br><input type="text" name="e-mail" size="35" value="пример: admin@zvirec.com">
```

Результат:

Введите свой e-mail:

пример: admin@zvirec.com

Теперь рассмотрим следующий элемент формы:

```
<form name="primer4" method="post" action="obrabotchik.php">
 Какой диск вы хотите получить?
<input name="disc" type="radio" value="cd" checked>
CD<br>
<input name="disc" type="radio" value="dvd">
DVD </form>
```

Результат:

Какой диск вы хотите получить?

C _{CD}

Данный тип элементов html формы называется радиопереключатель(переключает: либо одно значение, либо другое - два одновременно быть не может). Поэтому атрибут type имеет значение **type="radio"**. Раз мы выбераем между CD и DVD то радиопереключателя у нас два, поэтому два раза пользуемся элементом INPUT. Как видите у них одинаковое имя - **name="disc"** и разные значения **value**. Почему это так? давайте подумаем логически:

Нас интересует какой диск хочет получить клиент, CD или DVD. Поэтому у нас одинаковое значение имени **name="disc"** и разные значение (**value="cd"** и **value="dvd"**). Т.е. если мы выбираем первый вариант, то переменная **disc** примет значение **cd** а если второй- то **dvd**. Логично? по другому и быть не может...

Если вы хотите сделать, чтобы по умолчанию у Вас радиопереключатель был установлен на каком-либо варианте, то просто допишите атрибут-флаг **checked**(включен). Примечание:в радиопереключателе обязательно должен присутствовать атрибут **value** иначе ничего работать не будет.

Последний элемент в этом уроке:

```
<form name="primer5" method="post" action="obrabotchik.php">
Какие обучающие курсы вы хотите видеть на диске?
<input type="checkbox" name="fotoshop" value="yes" checked>
Курсы по Фотошопу <br>
<input type="checkbox" name="dreamweaver" value="yes">
```

Курсы по Adobe Dreamweaver
<input type="checkbox" name="php" value="yes">Курсы по PHP</br>

Результат:

Γ

Какие обучающие курсы вы хотите видить на диске?

Курсы по Фотошопу

Курсы по Adobe Dreamweaver

Курсы по РНР

Данный тип элементов html формы называется **checkbox** и отличается от радиопереключателя тем, что здесь можно выбрать несколько вариантов. Как видите type="**checkbox**" означает что тип элемента - Чекбокс, атрибут name нужен для того, чтобы обработчик мог идентифицировать данное поле и наконец value - определяет тот параметр, который будет отправлен при поставленной галочке.

В данном элементе, атрибут **value** не является обязательным, в отличие от радиопереключателя. Если мы его не поставим, то при поставленной галочке, как значение переменной в обработчик полетит текст который написан рядом с галочкой.

Остальные элементы формы мы разберем в следующем уроке. Пока поэксперементируйте и придумайте свои формы.

Урок 11: Заканчиваем разбираться с формами

Нам осталось разобрать три элемента формы. Давайте по-порядку:

<pre><form action="obrabotchik.php" method="post" name="primer6"> <pre></pre></form></pre>
<
<select name="dostavka" size="1"></select>
<option value="srochnaya"> Срочная </option>
<pre><option selected="" value="ne srochnaya"> He срочная </option></pre>
<option value="kurierom"> Курьером </option>

Результат:

Выберите способ доставки:

Не срочная

Базовым элементом здесь является **SELECT**(выбор). У него обязательно должен быть закрывающий тег! Внутри него содержатся элементы **OPTION** (опции выбора). Как вы поняли, без элементов option элемент select потеряет смысл, т.к. выбирать будет не из чего :)

Как видите, у элемента SELECT есть уже знакомый нам атрибут name, а у элементов OPTION тоже знакомый нам атрибут value (в элементе select это не обязательный атрибут). Они нужны для того, чтобы обработчик мог идентифицировать выбранный вариант.

Например мы выбираем способ доставки - Срочная. Тогда переменная dostavka примет значение srochnaya. Если б не было значения value="srochnaya" то переменная dostavka приняла бы значение Срочная (т.е. значение текста заключенного в элемент option). Атрибут **SIZE** задает количество одновременно видимых пунктов меню. В нашем случае это 1.

Атрибут флаг **SELECTED** означает что данное значение, будет выбрано по умолчанию. Можно также сделать, чтобы имелась возможность выбора нескольких пунктов меню при удержании клавиши Ctrl. Для этого используется атрибут-флаг MULTIPLE.

<form action="obrabotchik.php" method="post" name="primer7"></form>
Выберите способ доставки:
<select multiple="" name="dostavka" size="3"></select>
<pre><option value="srochnaya"> Срочная </option></pre>
<option value="ne srochnaya"> Не срочная </option>
<option value="kurierom"> Курьером </option>

Результат:

Выберите спо	соб доставки:
--------------	---------------

Срочная Не срочная Курьером

В теории это не сильно хорошо излагается, так что просто внимательней анализируйте приведенные примеры и будет Вам счастье :)

Давайте разберем следующий элемент:

<form action="obrabotchik.php" method="post" name="primer8"></form>
Введите адрес для доставки:
<textarea cols="45" name="adress" rows="5"></textarea>

Результат:

Введите адрес для доставки:	

Для создания больших текстовых полей используется элемент **TEXTAREA**, который даже так и переводится - текстовая площадь. У этого элемента обязательно наличие закрывающего тега.

Какие атрибуты у него есть? да знакомый нам атрибут **name** который поможет идентифицировать заполненное поле на сервере. Как видите здесь нигде нету параметра value, т.к. в качестве значения обработчику посылается текст, который мы введем в текстовое поле.

Атрибуты COLS и ROWS определяют ширину и высоту поля. К примеру cols="45" rows ="5" означает что ширина будет 45 колонок, а высота 5 рядов. Колонки и ряды естественно формируются из символов. Т.е. ширина 25 колонок, означает ширину в 25 символов, разобрались? думаю, что да.

И наконец мы добрались до кнопок:

<form action="obrabotchik.php" method="post" name="primer8"></form>
<input type="submit" value="Сделать заказ"/>
<input type="reset" value="Отмена"/>

Для создания кнопок используется уже знакомый нам элемент **INPUT**. Если необходимо создать кнопку, которая будет отправлять данные обработчику, то пишем

type="submit"(тип-отправить), а если хотите чтоб кнопка выполняла функцию сброса то **type="reset"**. Вот и все!

Атрибут value указывает на то, что будет написано на кнопке. Вас еще может напугать строка - это 6 пробелов подряд(это так называемые специальные символы html) я просто их поставил чтобы разделить кнопки! Все спец. символы есть у меня на сайте.

Теперь Мы всё умеем! следующий урок будет последним. там я дам вам обработчик для тренировки и расскажу как пользоваться.

Урок 12: Смотрим форму в действии

В этом уроке я, как и обещал, дам вам простейший обработчик и расскажу как протестировать форму.

Будем тестировать вот такую форму.

Введите ФИО:
Введите пароль:
Какой диск вы хотите получить?
C _{CD}
C _{DVD}
Какие обучающие курсы вы хотите видеть на диске?

Курсы по Фотошопу
Курсы по Adobe Dreamweaver
Курсы по РНР
Выберите способ доставки:
Срочная
Введите адрес для доставки:
Сделать заказ Отмена

Исходный html код формы:

```
<form name="zakaz_diska" method="post" action="obrabotchik.php">
Введите ФИO:<br>
<input type="text" name="fio" >
Введите пароль:<br>
<input type="password" name="pass">
 Какой диск вы хотите получить? 
<input name="disc" type="radio" value="CD" checked>
CD<br>
<input name="disc" type="radio" value="DVD">
DVD
<р> Какие обучающие курсы вы хотите видеть на диске? 
<input type="checkbox" name="photoshop" value ="Да" checked>
Курсы по Фотошопу
<br>
<input type="checkbox" name="dreamweaver" value ="Да">
Курсы по Adobe Dreamweaver
<br>
<input type="checkbox" name="php" value ="Да">
Курсы по РНР
```

Suffequate Cnocof доставки:	
Введите адрес для доставки : <textarea cols="45" name="adress" rows="5"></textarea>	
<input type="submit" value="Сделать заказ" width="20"/> <input type="reset" value="Отмена "/> 	

Пожалуйста разберитесь в исходном коде, все это Вы уже знаете и там нет ничего сложного. Как видите я сделал промежутки между различными типами элементов в исходном коде, чтобы Вам легче было разобраться.

Разобрались? теперь давайте протестируем Вашу форму, чтоб вы лучше поняли принцип.

Для этого Нам понадобится локальный сервер - это специальная программа, которая позволяет обрабатывать скрипты . Она нужна для отладки сайтов которые делаются на php(и других языках) у себя на локальном компьютере(а не на сервере в интернете). Поэтому не помешает научиться ее устанавливать если дальше собираетесь делать серьезные сайты.

<u>Скачайте эту</u> программу, запустите и следуйте инструкциям установщика. Единственное, по ходу установки нужно будет выбирать между первым и вторым режимом. По умолчанию идет первый, но мой Вам совет - используйте второй режим. После установки, у Вас на рабочем столе должны появится три ярлыка - Start Denwer, Stop Denwer, Restart Denwer(запуск, остановка и перезапуск сервера).

Запустите сервер (Start Denwer). После этого должен создаться виртуальный диск **Z**. Если Вы создавали во втором уроке свою первую страничку так, как я говорил, то у Вас должна быть папка obuchenie_html а внтури папка www, а внтури нее страничка index.htm в которую вы наверное уже вставили форму приведенную выше(не забудьте, что форма должна быть в теле документа!). Однако во втором уроке, мы работали с html, тут же мы будем работать с php, поэтому ситуация немного меняется. Чтобы php 100% работал, нужно обязательно нашу страничку с формой вставить в Z:\home\localhost\www т.е. на диск Z, в папку www, которая находится в папке localhost.

В итоге, у нас должно получиться следующее: папка obuchenie_html , внутри которой лежит файл с формой . Причем папка obuchenie_html лежит внутри папки www , которая в свою очередь лежит внутри папки localhost. Вот как выглядит путь :

Z:\home\localhost\www\obuchenie_html\index.html, где index.html файл с формой. Теперь осталось <u>скачать здесь архив с обработчиком</u>. Этот архив

самораспаковывающийся, т.е. его надо просто запустить и он сам распакуется. Внутри

него будет лежать файл обработчика obrabotchik.php. Его мы тоже копируем и помещаем в ту же папку где и наша страничка с формой.

Теперь запускаем локальный сервер (Start Denwer) и в строке браузера пишем: http://localhost/obuchenie_html . Появиться наша форма, которую можно заполнить и проверить.

Это самый простой обработчик, который просто выводит на экран данные заказа. Это чисто чтоб вы поняли как все происходит. После заполнения всех полей формы и нажатия кнопочки заказа, должно появиться что-то типа:

Сведения о заказе клиента :

Ф.И.О заказчика - Безруков А.Г. Пароль : 30061987 Тип диска - DVD

Хочет чтоб на диске были следующие курсы: Курсы по Фотошопу - Да Курсы по Adobe Dreameaver-НЕТ Курсы по РНР - Да

Выбрал способ доставки - Курьером В качестве домашнего адреса указал: Ул.Заднепровская, д.45 кв.123

Т.е. данные из одного документа полетели в другой и это наглядно видно. Так же само, можно сделать чтоб обработчик сразу отсылал данные о заказе на е-маил, или записывал их в базу данных и.т.д.

Надеюсь что Вам понятен основной принцип работы с формами!

Урок 13: Пару слов про голову документа

Все это время мы говорили про тело документа и уже научились делать свои страницы. Поговорим теперь про голову документа.

В первом уроке я говорил, что голова документа открывается тегом <HEAD> и закрывается тегом </HEAD> . Какие основные элементы содержатся в голове док-та? Ну во-первых это название документа - элемент TITLE . У этого элемента обязательный закрывающий тег. Это очень важный элемент.

Информация из элемента **TITLE** показывается в окне браузера(вверху слева), и поэтому это очень важный и информативный элемент для посетителей сайта. Если в TITLE написать например:

<title>Как создать сайт самому. Написать сайт самому. Сайтостроение</title>

Получим:



Тем более, первые 70 символов из TITLE показываются при поиске в Гугле и других поисковых системах. Обычно там используются ключевые слова по которым хотят раскручивать сайт.



Во-вторых это описание страницы сайта. Первые 170-200 символов видны при поиске в гугле поэтому оно тоже очень важно (на скриншоте выше описание это "Хотите создать и раскрутить сайт.....")

<META NAME="Description" CONTENT="Хотите создать и раскрутить сайт в сети интернет и.т.д."**>**

Здесь используется элемент **META** не имеющий закрывающего тега, с атрибутами **NAME**(определяет имя мета-записи в данном случае это описание - description) и **CONTENT** (определяет содержимое мета-записи).

В-третьих это набор ключевых слов. Сейчас это уже не актуально и нигде не используется. Поэтому можете даже и не применять. Раньше, эти слова учитывались поисковыми системами. Ну для общего развития смотрите как это делается:

<META NAME="Keywords" CONTENT="Сайт, интернет, htm, css">

Т.е все то же самое, только другое имя мета-записи **NAME="Keywords"** (от слова ключевые слова).

<head>

<title>Как создать свой сайт</title>

<meta name="Description" content="Если хотите научиться делать свои сайты, тогда вы попали по адресу!">

```
<meta name="Keywords" content="Сайт, интернет, html, css">
```

</head>

Ну, что еще сказать, также в голове документа прикрепляются таблицы стилей, но об этом уже поговорим в курсе CSS.

Урок 14: Завершение...

Вот Мы и закончили учебник по html . Надеюсь Вам все понятно, а если вдруг по ходу обучения возникают вопросы, пожалуйста задавайте их на форуме.

Как следующий шаг, Вам необходимо разобраться с технологией CSS, это очень интересная и нужная вещь. В этом нелегком деле Вам поможет мой учебник по css. Удачи и успехов Вам в обучении...

http://zvirec.com/html_sod.php

http://books.kulichki.com/data/html/html7/

http://on-line-teaching.com/html/index.html

http://ruseti.ru/html/