Вкратце, изучение HTML разбито на несколько частей:

A) Изучаем, что входит в <head></head>

B) Теперь переходим к <body></body:

1. Изучаем оформление текста, гиперссылок и картинок
2. Изучаем списки
3. Изучаем таблицы
4. Изучаем формы и кнопки

F) Изучаем правила оформления кода:

* используйте табуляцию, чтобы помочь визуализировать вложения элементов HTML
* вставляйте открывающие и закрывающие теги **блочных** элементов на отдельные строки
* пишите **строчные** элементы в одну строку (включая открывающие и закрывающие теги).
* Обязательно делайте комментарий, что это и для чего.

Евгений Попов учит, что **нужно обязательно указывать размер картинки в html коде (<img width=”” height=””>)**. Браузер, конечно, отобразит картинку и без этого указания, но браузеру так будет легче и быстрей определить место под картинку.

Хотя **<img>** относится и к строчным элементам, но занимает всю ширину экрана независимо от ширины картинки. Если мы добавим в тег **<img>** атрибут **align=”left”** или **align=”right”**, то данный контейнер поведёт себя как блочный элемент со свойством **float**. Т.е. картинка не полезет выше, чем она стоит в html файле, но картинка повиснет над блоками, идущими в html файле после **<img>**. Содержимое же этих блоков будет обтекать картинку. Всё совпадаёт с поведением блока **float**.

В **<img>** для регулировки размера изображения лучше давать только один из атрибутов **width** или **height**. Вторая сторона будет автоматически пропорциональной. Если задавать принудительно и **width**, и **height**, то может произойти диспропорциональное искажение.

<br> принудительный перенос применяют еще и для раздвижки элементов HTML по вертикали.

**!!!!** Для **id=** и **class=** нельзя задавать имена типа invest table. Правильно будет invest\_table или investTable. Вместо пробела используем нижнее подчёркивание, а лучше используем Camel-стиль. **Это оказалось очень важным**. У меня была сделана внутренняя якорная ссылка на id=”invest table”. Всё работало. Но после простой перезагрузки страницы в верхнем максимальном положении страница открывалась не на самом верху, а именно на этом id. Приходилось после перезагрузки перематывать страницу вверх. Мелочь, а неприятно. **Вставил нижнее подчёркивание, и проблема исчезла!** Также будет не очень корректно, если Id или class называть русскими словами или шифровать буквами и цифрами. Обозначения должны на английском языке точно выражать смысл этих селекторов (Google-переводчик в помощь). Не помешает и краткий комментарий о том, что это такое.

Для атрибута существует определённое правило записи. Атрибут вписывается всегда в первый (открывающий или единственный) тэг элемента html в следующем порядке: <**тэг + пробел + имя атрибута =”значение атрибута в кавычках”>** можно в одинарных кавычках ‘значение атрибута в кавычках’. Например: **<meta charset=’UTF-8’>. Пробел ставим только один раз.**

5) Когда пишем код цвета, то не забываем об английской клавиатуре: **#cc0000** и **#сс0000** – это разные вещи.

4) По умолчанию **HTML** отображает только один **пробел** между словами, сколько бы раз он ни был нажат. Чтобы **добавить** дополнительный **пробел**, напечатайте **&nbsp**; или **&#160**; . С помощью этого кода вы сможете отобразить символ, который называется **«неразрывный пробел»**. Неразрывный пробел не позволяет разорвать в этом месте строку. Не злоупотребляйте этим символом, иначе браузерам будет сложно вставлять разрывы строк аккуратным и понятным образом.

Существуют также и другие символы, которые сообщают браузеру, что нужно отобразить пробел. Различные браузеры по-разному их отображают, но, в отличие от **&nbsp;**, на разрывы строк они влиять не должны:

**a) &ensp;** – Названный так в честь блока «N» ручного типографского станка. Пробел длины «N» примерно равен длине двух стандартных пробелов.

**b) &emsp;** – Пробел длины «М» примерно равен длине четырех стандартных пробелов.

**6) Гипер Ссылки**

Основа Web это Гипер Текст, в котором есть гиперссылки. Гиперссылки бывают разных видов: Абсолютные, Относительные, Якорные, Относительно-якорные и Абсолютно-якорные.

**Абсолютная ссылка.**

Абсолютная ссылка в href имеет URL. <a href=”https://webref.ru/course/html-content/links”>Ссылка</a> При этом должен указываться весь URL полностью с протоколом.

**Относительная ссылка**

Относительные ссылки делают **относительно папки (каталога)**, в которой(ом) располагается файл HTML с ссылкой. Назовём условно наш html-документ **start.html**, каталог MainFolder, а файл, в который должна вести ссылка, назовём условно finish.html.

a) Итак, если **start.html** и finish.html находятся в одной MainFolder (в одном каталоге), то пишем в ссылке href=" finish.html".

b) Если finish.html находится в дочерней папке (Folder1) относительно MainFolder, то в href будем указывать адрес, начиная с имени дочерней папки через слэш. href=”folder1/ finish.html”

c) Если существует нисходящий ряд папок, вложенных в Folder1, то в href указываем через слэш весь этот ряд сверху вниз. href=”folder1/folder2/folder3/folder4/ finish.html”

d) Если наоборот finish.html находится в каталоге, который является родительским для MainFolder, то здесь нужно сделать один шаг вверх. Для этой цели используем **две точки+слэш** **«../»**. С помощью **«../»** мы поднимаемся вверх в родительскую папку. href=”../ finish.html”

e)Если существует множество родительских папок для MainFolder, то для каждого шага вверх на пути к finish.html указываем «../». href=”../../../../ finish.html”

f) Соответственно существует комбинированный вариант. href="../../../Files1/Files2/Files3/ finish.html" Поднимаемся на три папки вверх к общему предку, а затем спускаемся на три папки вниз к папке Files3, в которой и содержится файл finish.html.

g) **Краткое общее резюме.** Если **Start** и Finish находятся в одном каталоге (папке), то указываем просто название файла. Если Finish находится в другом каталоге, то для шага вверх используем **«../»**, а для шага вниз используем **«/»**. При необходимости комбинируем эти варианты. Относительные ссылки удобны тем, что при смене URL какого-либо изображения из интернета или при смене местоположения корневой папки на сервере, внутренние связи внутри корневого каталога не нарушаются. Главное — это тщательно заранее продумать семейное дерево папок внутри корневого каталога, чтобы **ничего потом не двигать и не нарушать связи.**

**Якорная ссылка.**

Если абсолютная и относительная ссылки просто перебрасывают пользователя на искомую веб-страницу, то якорная ссылка даёт возможность перебросить пользователя на нужное место какой-либо веб-страницы или на нужное место в рамках просматриваемой веб-страницы. Сам тэг <a></a> происходит от anchor.

Во-первых, кидается «якорь» в то место страницы, куда необходимо перебросить пользователя. Это универсальный атрибут id, например: id=”finish”. Имя задаётся в соответствии с правилами, освещёнными выше (3). Далее делается ссылка для перехода к якорю. Итак:

а) Якорь внутри страницы (якорная ссылка):

Это сам ЯКОРЬ: <p id=”finish”>Якорь</p>

А это уже ссылка на якорь: <a href=”#finish”>Перейти на Якорь</a>

b) Якорь на другой странице веб-сайта (**относительно-якорная ссылка**):

Это сам ЯКОРЬ: <p id=”finish”>Якорь</p>

Это ссылка на якорь: <a href=”folder/page.html#finish”>Перейти на Якорь</a>

c) Якорь на странице другого веб-сайта (**абсолютно-якорная ссылка**):

Это сам ЯКОРЬ: <p id=”finish”>Якорь</p>

Это ссылка на якорь: <a href=” https://webref.ru/course#finish”>Перейти на Якорь</a>

7) Можно таблицу вложить в таблицу. Для этого в элемент <td></td> вкладывается элемент <table></table>, в котором уже содержится таблица.

8) Если в таблице применяем colspan, rowspan, то нельзя забывать комментировать или удалять поглощённые ячейки. Иначе таблица будет кривой.

9) Тэг комментарий **<!-- -->** используется не только непосредственно для комментария, но и для того, чтобы временно скрыть «вырубить» ненужный участок кода.

**10) <Input>**

**<input>** самозакрывающийся тэг, с помощью которого создаются текстовые поля для ввода информации, переключатели, флажки, кнопки. Чем-то этот тег похож на <img>. Это тоже своеобразная картинка, только предназначенная для ввода или управления информацией. Итак:

a) <input type='text' name='userName' placeholder='Введите Ваше имя не короче 8 и не длиннее 20 знаков' size='60' minlength='8' maxlength='20'>

Перед нами окошко для ввода текстовой информации без защиты, т.е. вводимая информация при значении атрибута type=’text’ будет видна посторонним глазам. У каждого такого окошка должно быть своё индивидуальное имя name=’’. Это важно для обработчика. Размер-длина по умолчанию 20 знаков моноширинного шрифта. С помощью атрибута size=’’ этим параметром можно управлять. Впрочем, лучше длину задавать через CSS. Атрибут placeholder=’’ задаёт в окошко текст, который исчезаёт при начале ввода. Данный атрибут хорошо использовать для подсказки юзеру, например, о количестве максимальных и минимальных знаков. Атрибут minlength=’’ задаёт минимальное количество знаков. При меньшем количестве знаков браузер будет сигнализировать. Атрибут maxlength=’’ задаёт максимальное количество знаков. Большее количество знаков, просто, нельзя будет ввести. Таким образом, главный атрибут здесь это type=’text’. Именно он задаёт тип тега <input>.

b) <input type='password' name='userPassword' placeholder='Введите Ваш пароль не короче 8 и не длиннее 15 знаков' size='60' minlength='8' maxlength='15'>

Перед нами окошко ввода текстовой информации с защитой от посторонних глаз. Так можно защищать любую информацию (например, E-mail), но особенно это касается защиты паролей. Поэтому данное окошко создаётся с помощью атрибута type=’password’. Всё остальное аналогично форме а).

c) <input type=’radio’ name=’country’ value=’Russia’>

<input type=’radio’ name=’country’ value=’France’ checked>

Перед нами переключатель, или радиокнопки. Кнопки являются альтернативными: мы нажимаем одну, а другая отжимается автоматически. Для этого у кнопок строго должно быть одно имя, задаваемое атрибутом name=’’. Если имена будут разные, то альтернативности не будет. Атрибут value=’’ в данном случае **(это важно!)** задаёт значение кнопки, которое будет передано обработчику. Атрибут type=’radio’ задаёт тип кнопки. Если мы хотим задать выбор по умолчанию, то в нужный переключатель вставляем атрибут checked.

d) <input type=’checkbox’ name=’London’>

<input type=’checkbox’ name=’Moscow’ checked>

Перед нами чекбоксы, или флажки. Если две и больше радиокнопок – это переключатель, то один чекбокс — это уже включатель\выключатель. Тип определяется атрибутом type=’checkbox’. Имена должны быть разными, это важно для обработчика. Если мы хотим задать выбор по умолчанию, то в нужный чекбокс вставляем атрибут checked.

e) <input type='reset' name='reset' value='Очистить формы'>

Перед нами кнопка «Сброс». Она производит очистку всех форм и возвращение к настройкам по умолчанию в этих формах. Атрибут type=’reset’ задаёт тип кнопки. Атрибут value=’’ здесь даёт отображаемое наименование. Если в радиокнопке этот атрибут даёт значение для обработчика, то здесь даёт видимое наименование. Мы можем не давать value=’’, тогда на кнопке выйдет название по умолчанию «Сброс».

f) <input type='submit' name='submit' value='OK'>

Перед нами кнопка «Отправить запрос». Эта кнопка производит отправку обработчику данных, которые введены или выбраны в формах. Тип кнопки задаёт атрибут type=’submit’. Как и в кнопке «Сброс» здесь value=’’ задаёт видимое наименование кнопки. Если value=’’ не указать, то по умолчанию на кнопке будет написано «Отправить запрос».

g) <input type=’image’ src=’images/pic1.png’ alt=’Submit’ name=’Submit’>

Перед нами ещё одна разновидность кнопки «Отправить запрос». В атрибуте указано type=’image’, т.е. кнопка создаётся с помощью картинки. Нужно понимать, что данная картиночная кнопка работает только как «Submit» и больше никак.

h) <input type='button' name='button' value='ВЕРНУТЬСЯ ОБРАТНО ВВЕРХ'>

Перед нами просто кнопка. Атрибут value=’’ задаёт видимое наименование кнопки. Эту кнопку можно использовать для различных целей. Например, можно сделать кнопку «Вернуться к началу страницы». Для этого кнопку помещаем в контейнер <a></a>. Делаем якорную ссылку на начало страницы. Подчёркивание ссылки можно убрать с помощью CSS.

<a href='#top'>

<input type='button' name='button' value='ВЕРНУТЬСЯ ОБРАТНО ВВЕРХ'>

</a>

<style>

a { text-decoration: none; /\* Убираем подчеркивание у всех ссылок \*/

}

</style>