

1. Основные понятия в веб-разработке

Что представляет собой веб-страница. Виды сайтов. Процесс разработки сайта. Что такое гипертекст, теги и атрибуты. Подготовка рабочего места. Структура HTML-документа. Основные теги оформления текста. Простой пример HTML странички. Пример сложной веб-страницы. Горячие клавиши для перемещения по документам.

1.1. Как работает интернет

Прежде всего давайте ответим на вопрос: что такое интернет? За ответом на этот вопрос обратимся к Википедии, и в этом справочнике найдем следующее определение:

Интернёт (англ. Internet) — всемирная система объединённых компьютерных сетей, построенная на использовании протокола TCP/IP и маршрутизации пакетов данных. Интернет образует глобальное информационное пространство, служит физической основой для Всемирной паутины (WWW, World Wide Web) и множества других систем (протоколов) передачи данных. Часто упоминается как «Всемирная сеть» и «Глобальная сеть», в обиходе иногда употребляют сокращённое наименование «Инёт». Другими словами, Интернет состоит из множества домашних и корпоративных сетей, принадлежащих различным пользователям, компаниям и предприятиям, работающих по самым разнообразным протоколам, связанных между собой различными линиями связи, которые могут передавать данные по телефонным проводам, оптоволокну, через спутники и радиомодемы.

Интернет — это множество компьютеров по всему миру, объединённых в единую сеть, которые постоянно обмениваются какой-либо информацией. Структура Интернет напоминает паутину, в узлах которой находятся компьютеры, связанные между собой линиями связи. Каждый компьютер в свою очередь имеет свой уникальный адрес, называемый IP-адресом, который обязательно должен быть уникальным, так же, как и любого объекта в реальной жизни есть свой уникальный адрес. Существуют два типа IP-адреса:

- постоянные, закреплённые за определённым компьютером.
- динамические, которые присваиваются в тот момент, когда пользователь соединяется с интернетом.

Структура IP-адреса устроена таким образом, что мы можем узнать, в какой стране и в каком городе находится компьютер пользователя. Таким образом, например, при настройке объявлений по контекстной рекламе, можно задавать регионы показа нашего объявления.

Пример IP-адреса: 192.168.1.2

Для того, чтобы узнать свой IP-адрес, нужно зайти в «панель управления» — «сетевые подключения» — выбрать «подключение по локальной сети» и перейти на вкладку «поддержка».

Для того, чтобы не запоминать сложные IP-адреса, были придуманы так называемые доменные имена.

Доменное имя — уникальное имя, которое данный поставщик услуг избрал себе для идентификации, например, mail.ru или google.com.

Доменное имя может иметь несколько уровней. Домен 1 уровня обычно определяет страну местоположения сервера (ru — Россия; ua — Украина; uk — Великобритания; de — Германия) или вид организации (com — коммерческие организации; edu — научные и учебные организации; gov — правительственные учреждения; org — некоммерческие организации). С недавнего времени стало возможным использование русскоязычных доменов (рф).

1.2. Браузеры

Браузер — это программа, с помощью которой вы имеете возможность просматривать любые сайты.

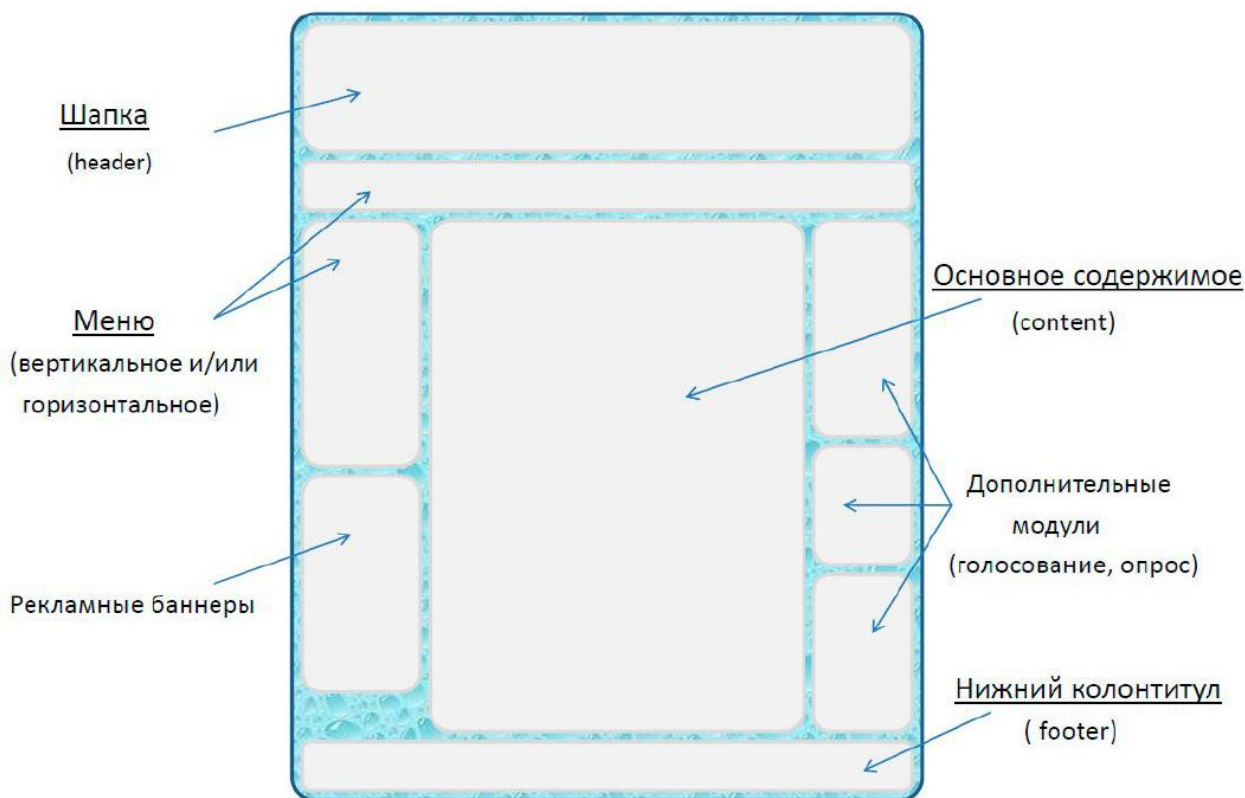
То есть браузер является проводником между всемирной сетью Интернет и пользователем. Именно он позволяет получать доступ ко всему множеству доступной в Интернете информации.

Существует большое количество браузеров. Большинство из них практически одинаково отображают информацию, за исключением Internet Explorer, который некоторые теги и свойства стилей отображает иначе — к примеру, воспринимает иначе ширину блоков. Эти особенности необходимо учитывать при верстке, но об этом подробнее в конце курса.

1.3. Что представляет собой веб-страница

Веб-страница сайта состоит из различных блоков:

- Шапка (header), в которой могут размещаться: логотип компании, название сайта, телефоны организации и проч.
- Вертикальная и/или горизонтальная навигация по сайту (меню).
- Основное содержимое (content).
- Рекламные баннеры и ссылки на другие сайты.
- Дополнительные модули, например, голосование, опросы, корзина покупателя.
- Нижний колонтитул (footer), где размещается дополнительная информация: автор сайта, счетчики посещаемости.



1.4. Виды сайтов

По содержанию

Статические — содержимое подготавливается заранее и выдается пользователю в том виде, в котором хранится на сервере.

Динамические — содержимое генерируется при помощи серверных языков программирования.

По схеме представления информации

Коммерческие — сайты компаний, интернет-магазины и т.д.

Информационные — доносят до пользователя какую-либо информацию.

Веб-сервисы (порталы) — поисковые системы, электронная почта, форумы, социальные сети.

По размеру

Фиксированной ширины — задается фиксированная ширина сайта (в настоящее время ширина обычно составляет 1000 пикселей, т.к. разрешение современных мониторов по ширине начинается от 1024 пикселей). Тогда все блоки сайта будут выглядеть так, как и было задумано и не будут изменять размер в зависимости от разрешения монитора или размера экрана браузера.

«Резиновые» — ширина строго не задается, а изменяет размер в зависимости от размера экрана браузера или разрешения монитора.

Адаптивная вёрстка/тип макета — дизайн, который подстраивается (адаптируется) под размер экрана, в том числе может происходить перестройка блоков с одного места на другое, или их замена блоками, отображаемыми только при определённом разрешении.

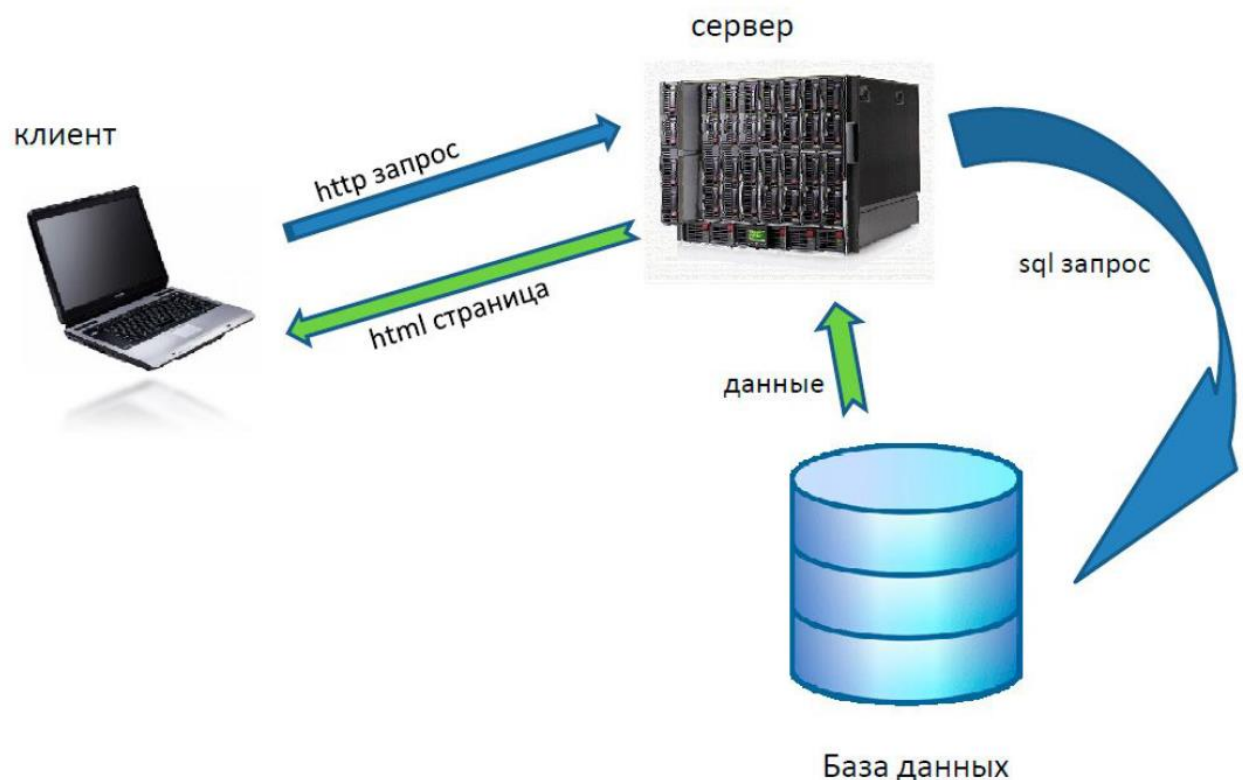
1.5. Протоколы передачи данных

Взаимодействие компьютеров, серверов, маршрутизаторов, коммутаторов определяется протоколами. Каждый протокол — четкий и определенный набор правил и соглашений, предписывающий, каким образом производится обмен и обработка информации.

Наиболее известные протоколы, используемые в сети Интернет:

- HTTP (HyperText Transfer Protocol) — это протокол передачи гипертекста. Протокол HTTP используется при пересылке Web-страниц с одного компьютера на другой.
- HTTPS (HyperText Transfer Protocol Secure) — это тоже протокол для передачи гипертекста, но использует дополнительное шифрование данных, для более безопасной передачи информации.
- FTP (File Transfer Protocol) — это протокол передачи файлов со специального файлового сервера на компьютер пользователя. FTP дает возможность абоненту обмениваться двоичными и текстовыми файлами с любым компьютером сети. Установив связь с удаленным компьютером, пользователь может скопировать файл с удаленного компьютера на свой или скопировать файл со своего компьютера на удаленный.

1.6. Схема HTTP-запроса страницы



Пользователь набирает в браузере адрес нужного ему сайта, после чего посылается HTTP-запрос серверу. Сервер определяет тип страницы по расширению запрашиваемого файла. Если расширение .html, то это страница статическая, и он сразу формирует html-страничку и отправляет ее обратно пользователю. Если расширение, к примеру .php, то в этом случае, страница уже динамическая, и сервер, в этом случае, подключает интерпретатор php, который в свою очередь может обратиться в базу данных и получить оттуда необходимые данные.

Результат — HTML-код, который веб-сервер посылает обратно на компьютер пользователя. Сгенерированный HTML-код веб-сервер упаковывает в HTTP-пакет, который и передается по сети обратно клиенту. Полученный HTTP-ответ попадает в браузер клиента, который извлекает из него HTML-код и генерирует на его основе графическое представление запрошенной страницы.

Простым языком: если мы обратились к html файлу (.html), то браузеру этого документа достаточно и Apache в случае такого запроса возвращает нам html разметку. Если используется другое расширение файлов, например PHP (.php), в таком случае в данном файле может содержаться программный код, который должен обработан (выполнен) интерпретатором языка. На выходе php-интерпретатор должен нам выдать html-код, потому что именно его понимает браузер.

1.7. Процесс разработки сайта



На данной схеме представлен процесс разработки сайта на примере веб-студии, где определенные функции выполняет определенный человек. Когда заказчик приходит со своей идеей сайта в веб-студию с ним начинает работать менеджер проекта. Очень часто бывает, что идея то у заказчика есть, но она настолько размыта, что тяжело представить на этом этапе, что в итоге хочет получить клиент.

В разных организациях используется свой подход к выяснению потребностей заказчика. Один из вариантов — специальная анкета, после составления которой можно уже представить, как будет выглядеть и каким функционалом обладать будущий сайт.

Далее менеджер составляет два технических задания, одно дизайнеру, на составление макета сайта (дизайна), а другое — программисту, с указанием какой функционал необходимо реализовать в этом проекте.

После этого верстальщику отправляется макет сайта от дизайнера, и различные модули от программиста, и после того, как сайт будет сверстан, копирайтер наполняет его контентом, т.е. заполняет товарами, пишет статьи и т.д. И в это же время seo-специалист начинает заниматься продвижением сайта.

1.8. Гипертекст

Гипертекст — термин, введенный Тедом Нельсоном в 1965 году для обозначения «текста ветвящегося или выполняющего действия по запросу». Обычно гипертекст представляется набором текстов, содержащих узлы перехода между ними, которые позволяют избирать читаемые сведения или последовательность чтения.

Общеизвестным и ярко выраженным примером гипертекста служат веб-страницы — документы HTML (язык разметки гипертекста), размещённые в Сети. В более широком понимании термина, гипертекстом является любая повесть, словарь или энциклопедия, где встречаются отсылки к другим частям данного текста, имеющие отношения к данному термину. В компьютерной терминологии, гипертекст — текст, сформированный с помощью языка разметки, потенциально содержащий в себе гиперссылки.

Теги и атрибуты

Язык HTML — язык тегов. Теги описывают структуру HTML-документа. Теги оформляются угловыми скобками <имя тега>, между которыми прописывается имя тега. Теги HTML документа предназначены для управления конструкциями разметки — заголовками, абзацами, списками, таблицами и картинками.

Теги бывают парные и одиночные.

Парные теги сначала открываются, потом закрываются. Все, что описывают парные теги, находится внутри них. Так, тег делает текст полужирным. Вот как это выглядит:

 Какое прекрасное утро

Одиночные теги сами по себе, и как правило не меняют контент вокруг себя. Пример:

 — тег переноса строки

<hr /> — тег горизонтальной линии

Атрибуты тегов определяют какие-либо дополнительные, уточняющие параметры того или иного тега.

Вы можете указать внутри тега столько атрибутов, сколько вам необходимо. Атрибуты разделяются пробелами.

<название_тега атрибут1="значение_атрибута1"> текст внутри тега </ название_тега >

Пример:

 Контакты

1.9. Структура HTML документа

Каждая страница в HTML-документе состоит из трех обязательных элементов:

1. Тип документа
2. Раздел <head></head> с технической информацией о странице: заголовок, описание, ключевые слова для поисковых машин, кодировка. Введенная в нем информация в основном не отображается в окне браузера, однако содержит данные, которые указывают браузеру, как следует обрабатывать страницу.
3. Раздел <body></body>, в котором уже располагаются все элементы, который видит пользователь. Именно с этим разделом мы, в основном, и будем работать.

Элемент <!DOCTYPE> предназначен для указания типа текущего документа — DTD (document type definition, описание типа документа). Это необходимо, чтобы браузер понимал, как следует интерпретировать текущую веб-страницу, поскольку HTML существует в нескольких версиях, кроме того, имеется XHTML (EXtensible HyperText Markup Language, расширенный язык разметки гипертекста), похожий на HTML, но различающийся с ним по синтаксису. Чтобы браузер «не

путался» и понимал, согласно какому стандарту отображать веб-страницу и необходимо в первой строке кода задавать `<!DOCTYPE>`.

Элементы, находящиеся внутри тега `<html>`, образуют дерево документа, так называемую объектную модель документа, DOM (document object model). При этом элемент `<html>` является корневым элементом.

Пример структуры HTML 5

```
<!doctype html>
<html>
<head>
<title> Hello HTML </title>
<meta charset = "UTF-8">
</head>
<body>
...
</body>
</html>
```

1.10. Основные теги оформления текста

Заголовки

Так же, как и в газетах и журналах, в HTML-странице любая статья или новость должна начинаться с заголовка. Предусмотрено 6 уровней заголовков, первый из которых будет отображен самым крупным шрифтом, а далее, чем выше уровень, тем размер шрифта будет уменьшаться.

Для отображения заголовков существует тег `<h>`, и указывается цифра от 1 до 6, которая соответствует уровню заголовка. Тег заголовка — парный тег, не забудьте его закрыть.

`<h1>Заголовок первого уровня</h1>`

`<h2>Заголовок второго уровня</h2>`

`<h3>Заголовок третьего уровня</h3>`

`<h4>Заголовок четвертого уровня</h4>`

`<h5>Заголовок пятого уровня</h5>`

`<h6>Заголовок шестого уровня</h6>`

Параграфы

Далее, после заголовка обычно следует какой-нибудь текст, который необходимо заключать в параграфы, или иначе сказать, абзацы. При составлении документа выделяйте блоки текста в отдельные параграфы, как это сделано в книгах, в противном случае, может получиться сплошной текст, который очень трудно будет читать посетителю вашего сайта.

В HTML для параграфов используется парный тег `<p>`, и внутри него помещается тот текст, который нужно отобразить. Пример:

`<p> Здесь мы напишем первый параграф </p>`

Теги выделения текста

Иногда необходимо выделить слово, словосочетание, предложение или целый участок текста, чтобы привлечь внимание читателя или поискового робота. А для того, чтобы это сделать, можно выделить фрагмент жирным, курсивом, или подчеркнуть.

Аккуратнее с подчеркиванием текста: принято по стандартам, что подчеркнуты гиперссылки, и многие пользователи уже настолько к этому привыкли, что когда видят подчеркнутый текст, то им хочется «кликнуть» по этому участку текста, ожидая перейти на другую страницу.

Теги **** и **** внешне делают одно и то же, делая текст полужирным. Разница в том, что **** указывает на важность текста, а **** просто делает текст полужирным. Так, текст в тегах **** устройство для чтения текста вслух будет выделять интонацией, а **** — нет.

Аналогично с тегами **** и **<i>**, делающими текст курсивным. **** указывает на важность текста, а **<i>** нет.

<small> Уменьшает размер шрифта на единицу по отношению к обычному тексту.

<sub> Используется для создания нижних индексов. Сдвигает текст ниже уровня строки, уменьшая его размер.

<sup> Используется для создания степеней и верхних индексов. Сдвигает текст выше уровня строки, уменьшая его размер.

<ins> Выделяет текст в новой версии документа, подчёркивая его.

**** Перечёркивает текст. Используется для выделения текста, удаленного из документа.

<code> Служит для выделения фрагментов программного кода. Отображается моноширинным шрифтом.

<pre> Позволяет вывести текст на экран, сохранив изначальное форматирование. Пробелы и переносы строк при этом не удаляются.

<q> Используется для выделения коротких цитат. Браузерами заключается в кавычки.

Вложенные теги

Очень часто при верстке веб-страниц необходимо вкладывать одни теги в другие. И эта вложенность может достигать нескольких уровней. Поэтому следует запомнить одно простое правило вложенных тегов: закрывать теги необходимо именно в той последовательности в которой вы их открывали.

Первый открытый тег закрывается последним.

<p> Использовать вложенность тегов

**** легко

<i> и просто **</i>**

</p>

Тег **<i>** был открыт последним — закрываем его первым, а далее уже все остальные теги по очереди.

1.11. Необходимые инструменты

Текстовый редактор

Для того, чтобы начать создавать свои веб-странички, нам необходим текстовый редактор.

Можно воспользоваться встроенным в Windows блокнотом, но для начала удобнее использовать более функциональными текстовыми редакторами Sublime text, Notepad++ или Brackets.

Браузер

Выберите для себя наиболее удобный вам браузер, только на начальном этапе НЕ используйте Internet Explorer. Лучше использовать Google Chrome.

Графический редактор

Photoshop или Gimp — для «нарезки» макета сайта и подбора цветов.

1.12. Горячие клавиши

Alt + tab – перемещение между открытыми окнами.

Ctrl + z – отменить действие.

Ctrl + y – вернуть действие.

Ctrl + x – вырезать.

Ctrl + c – копировать.

Ctrl + v – вставить.

Ctrl + s – сохранить.

Ctrl + a – выделить все.

Ctrl + tab – перемещение м/у открытыми вкладками.

F5 – обновить страницу.

F2 – переименовать.

Shift + Tab – вернуть одну табуляцию назад.

Shift + home – выделить с указанного места до начала строки.

Shift + end – выделить с указанного места до конца строки.

Shift + стрелка – при зажатом shift выделять в указанную стрелками сторону.

1.13. Практика

1. Создаем на любом из жестких дисков новую папку.
2. Далее заходим в эту папку, щелкаем правой кнопкой мыши, выбираем пункт «создать текстовый документ» с расширением «.txt».
3. Переименовываем этот файл, даем ему любое имя, а расширение меняем на «.html»
4. Щелкаем по файлу правой кнопкой мыши, выбираем пункт «открыть с помощью», выбираем текстовый редактор и открываем.
5. Напишем теперь первое приветствие миру. Переходим в текстовый редактор и набираем:
`<h1>Hello world!</h1> <p>I'm here</p>`
6. Чтобы посмотреть результат в браузере — щелкаем по этому файлу.

2. Основы HTML

Спецсимволы. Комментарии в HTML. Гиперссылки. Типы изображений для web. Загрузка изображений на страницу, Списки, Формы и их элементы.

2.1. Спецсимволы

Спецсимволы — это символы, которых нет на клавиатуре. Например, знак копирайта ©. Можно найти в гугле и вставить, но не в каждом редакторе он будет корректно отображаться. Спецсимволы добавляются с помощью амперсанда &. Знак копирайта будет выглядеть как ©. После спецсимвола ставится точка с запятой.

Код в HTML	Внешний вид	Описание
 		неразрывный пробел
©	©	знак copyright
" и »	“ и »	двойные кавычки
< и >	< и >	символы
′	'	одиночный штрих

Спецсимволы могут быть разные, например, кавычки или неразрывный пробел. Кавычки на клавиатуре есть, но в редакторе ими оформляются ссылки, поэтому для отображения их на странице тоже надо использовать спецсимволы. Неразрывные пробелы надо использовать для нормального отображения текста на всех устройствах и браузерах, чтобы предлоги были рядом со словами, к которым они относятся (можно продемонстрировать, изменяя ширину окна). Для вставки неразрывного пробела надо использовать спецсимвол . Чтобы не добавлять неразрывные пробелы вручную, существует инструмент типограф. Их достаточно много в интернете, напр. типограф Лебедева (<https://www.artlebedev.ru/typograf/>). Он позволяет обработать весь текст и проставить все спецсимволы, а не только пробелы.

2.2. Комментарии

Комментарии используются для добавления информации в тексте, которая не будет отображаться на странице. Например, примечания/пометки для других разработчиков, или если блок информации нужно скрыть, не удаляя. Комментарии оформляются с помощью тега `<!-- текст комментария -->`.

(ctrl + /)

Пример:

`<!-- Текст, который не будет отображаться на сайте -->`

`<p> Текст основной части сайта </p>`

2.3. Гиперссылки.

Гиперссылка — активный элемент (текст, изображение, кнопка и т.п.) одного документа, при нажатии на который происходит переход к другому документу или к его элементу. Документы, содержащие гиперссылки, называются гипертекстовыми. Гиперссылка может быть добавлена к любому элементу гипертекстового документа и обычно выделяется графически.

В HTML-документах текстовые ссылки по умолчанию выделяются синим цветом, при наведении на них курсором мыши в окне браузера изменяются, например, меняют цвет или выделяются подчеркиванием. Посещенная ранее ссылка обычно выделяется цветом, отличным от цвета не посещенной ссылки.

Тег ссылки называется <a> и у нее есть обязательный атрибут href , в значении которого мы указываем на тот файл, на то место в документе, или на ту страницу другого сайта, куда хотим перейти. В содержимом этого тега необходимо написать тот текст, который увидит пользователь.

2.4. Виды ссылок

Относительные ссылки используются для перемещения внутри документа или сайта.

```
<a href = "file_name.html" >
```

[текст ссылки, который видит пользователь](#)

```
</a>
```

Абсолютные ссылки используются для перехода на страницы внешнего сайта, для этого в значении атрибута href нужно указать полный путь до той страницы, на которую мы хотим перейти, включая тип протокола.

```
<a href = " http://yandex.ru " target="_blank" >
```

[страница yandex.ru откроется в новой вкладке](#)

```
</a>
```

У тега ссылки существует атрибут target . Если указать target="_blank" , то страница откроется в новой вкладке браузера. Раньше это значение официально не поддерживалось в HTML, но люди все равно использовали его, поскольку оно работало. Теперь в HTML5 оно полностью поддерживается.

Когда использовать target="_blank"

- Если необходимо открыть сторонний ресурс и не потерять страничку вашего сайта
- Если пользователь работает над чем-то, что может быть потеряно если текущая страница изменится.

Когда не использовать target="_blank"

- В меню (иначе пункты меню будут открываться в новых вкладках)
- Из-за личных предпочтений
- Если клиент думает, что так будет лучше

Якоря в гиперссылках

При помощи гиперссылок можно перемещаться не только между страницами документа, но и внутри данной страницы. Это бывает очень полезно, когда ваша статья состоит из нескольких разделов и имеет довольно объемное содержание.

В этом случае разумно будет использовать так называемые якоря, чтобы быстро переходить в интересующий нас раздел. В том элементе, на который нам следует перейти, нужно определить уникальный идентификатор.

Для этого существует атрибут id, который можно определять практически для любого тега. Название этого атрибута может быть любым и обязательно должно быть уникальным. В значении атрибута href тега-ссылки ставится символ решетки, после которого указывается имя того идентификатора, к которому необходимо обратиться.

```
<a href = "#p10" > Прочитать 10 параграф </a>
```

```
<p id = "p1" > Текст параграфа №1 </p>
```

```
<p id = "p2" > Текст параграфа №2 </p>
```

...

<p id = "p10" > Вот и долгожданный параграф №10 </p>

2.5. Типы изображений

Цель – понять, когда и какие типы графических изображений следует использовать. Главная задача – чтобы страница грузилась максимально быстро без ущерба для качества сайта.

Формат jpg

Плюсы:

- Поддерживает 16 млн цветов
- Мало весит
- Можно управлять качеством изображения при сохранении (например, поставить 80% - разница не будет заметна)

Минусы:

- При сжатии теряется качество изображения
- Не поддерживает прозрачность (картинка всегда прямоугольная и занимает всю площадь)

Формат gif

Плюсы:

- Поддерживает прозрачность
- Поддерживает анимацию
- При сжатии в качестве не теряет

Минус:

- Количество цветов 256

Формат png

Растровый графический формат PNG, набирающий всё большую популярность, появился в 1995 году как замена GIF. Формат PNG в свою очередь подразделяется на форматы PNG-8 (аналог формата gif, за исключением того, что PNG-8 не поддерживает анимацию) и PNG-24.

Еще существует формат PNG-32 — в этом формате растровой графики для одного пикселя возможно использовать четыре байта информации. Три отвечают за формирование полноцветного изображения, а четвертый байт выделяется отдельно для формирования альфа-канала, позволяющего использовать PNG-32 для получения картинок с прозрачным фоном.

PNG – 8

Плюсы:

- Поддерживает прозрачность
- При сжатии не теряет в качестве

Минусы:

- Кол-во цветов до 256

PNG – 24

Плюсы:

- При сжатии не теряет в качестве
- 16 млн цветов
- Плавный переход от прозрачной области к цветной (прозрачный фон)

Минусы:

- Большой размер файла

При одинаковом разрешении 150x150 px размер файла составит:

- jpg – 12 kB
- gif – 45,5 kB
- png – 200,5 kB

Конечно, при современных скоростях доступа в интернет это может показаться несущественным, но ситуации могут быть разные. Загрузка страницы больше 3-х секунд считается правилом дурного тона, и пользователь может потерять интерес к нашему ресурсу.

SVG

SVG — формат векторной графики, файлы SVG можно читать и редактировать (при наличии некоторых навыков) при помощи обычных текстовых редакторов. Существует возможность увеличить любую часть изображения SVG без потери качества.

Широко доступно использование растровой графики в SVG документах. Имеется возможность вставлять элементы с изображениями в форматах PNG, GIF или JPG. Текст в графике SVG является текстом, а не изображением, поэтому его можно выделять и копировать, он индексируется поисковыми машинами, не нужно создавать дополнительные метафайлы для поисковых роботов.

Плюсы:

- Небольшие размеры файлов, отличное сжатие;
- Возможно увеличить любую часть изображения SVG без потери качества

Минусы:

- Большой размер файла
- Сложность использования

Из рассмотренных сильных и слабых сторон рассмотренных форматов изображений можно сделать вывод. Для сохранения фотографий с четкими краями лучше всего подходит формат jpeg, так как в нем размер файла получается небольшим.

Для анимированных изображений единственный вариант — использовать формат gif, а для сохранения качественных изображений, в которых используется прозрачность лучше всего подойдет формат png-24. Для векторной графики идеально подойдет svg.

Загрузка изображений на страницу

Для отображения изображений на странице существует тег ``, который является одиночным тегом. У этого тега есть обязательный атрибут `src`, в значении которого мы указываем путь к изображению. Этот путь может быть как относительным так и абсолютным.

```
<img src = "img/my_foto.jpg"
```

```
alt = "это моя фотография"
```

```
title = "это моя фотография"
```

```
width = "300" height = "300"
```

```
/>
```

У тега `` можно указать дополнительные атрибуты.

Атрибут alt означает альтернативный текст. Используется он для того, чтобы: во-первых, если картинка не загрузится на страницу, то пользователь увидит тот текст, который вы указали в этом атрибуте, и во-вторых, альтернативный текст необходим при продвижении сайта, в нем можно указать ключевые слова, и тогда это изображение будет участвовать в поиске по картинкам.

Следующий атрибут — title, который является универсальным, и его можно использовать практически для любого тега. То, что вы напишите в его значении, будет выводиться в виде всплывающей подсказки.

Любому изображению можно задать ширину и высоту, указав эти значения в атрибутах width и height соответственно. При помощи этих атрибутов изображение лучше не увеличивать, а то получится очень плохое качество. Если задать ширину и высоту изображению, то браузер при загрузке страницы будет сразу выделять заданную область под картинку.

Всегда храните все изображения в отдельной папке, тогда вы будете знать, где искать то или иное изображение.

2.6. Списки

Списки в html-документах бывают двух видов: маркированные и нумерованные. Отличаются они тем, что у маркированных элемент списка начинается с маркеров, а у нумерованного с цифр или букв.

Структура маркированного списка

```
<ul>
  <li> Первый элемент списка </li>
  <li> Второй элемент списка </li>
  <li> Третий элемент списка </li>
</ul>
```

Структура нумерованного списка

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

Как видно из примеров структура обоих типов списков одинаковая, и единственное различие в том, что в случае с маркированным списком, контейнер будет называться , а в нумерованном . В нумерованном списке иногда может потребоваться начинать список с какой-нибудь определенной цифры или буквы. Для этого существует атрибут start, в значении которого указывается, с какого элемента будет начинаться список. В списках есть возможность менять вид маркеров, или арабские цифры изменить, например, на римские, либо вообще убрать маркеры.

Структура вложенного списка

```
<ol>
  <li> Первый элемент списка </li>
  <li> Второй элемент списка
    <ul>
      <li> Первый элемент вложенного списка </li>
      <li> Второй элемент вложенного списка </li>
      <li> Третий элемент вложенного списка </li>
    </ul>
  </li>
  <li> Третий элемент списка </li>
```


Вложенные списки предназначены для организации сложной иерархической структуры текста, обычно таких документов, как юридические или технические. Также вложенные списки используются для создания многоуровневых меню и навигации по сайту.

2.7. Формы и их элементы

Формы необходимы для того, чтобы было возможно получать от пользователя информацию, отправлять эти данные на сервер, а там уже их обрабатывать. Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

Формы для ввода задаются тегом `input`. По умолчанию стоит тип текст (ввод текста).

Допускается внутри контейнера `<form>` помещать другие теги, при этом сама форма никак не отображается на веб-странице, видны только ее элементы и результаты вложенных тегов.

Структура форм

Все элементы форм должны обязательно находиться внутри контейнера `<form>`. В этом теге могут находиться атрибуты, необходимые для правильной работы формы.

Элемент `<fieldset>` предназначен для группирования элементов формы. Такая группировка облегчает работу с формами, содержащими большое число данных. Например, один блок может быть предназначен для ввода текстовой информации, а другой — для флажков. Внутри тега `<legend>` поместить название данной формы. Тогда браузер выделит эту часть рамкой, в которой будет находиться название.

Элемент `<legend>` не обязателен, но если присутствует, должен идти сразу же после тега `<fieldset>`. Другие теги или текст перед `<legend>` недопустимы. Внутри `<legend>` можно использовать теги форматирования вроде ``, `<i>`

```
<form action="URL" >
<fieldset>
<legend> Контактные данные </legend>
...
</fieldset>
</form>
```

Элементы форм

В html существует три основных тега элементов форм

1. `<textarea>` используется для того, чтобы сформировать многострочное поле ввода для большого количества информации. С помощью атрибута `cols` можно задать ширину этого поля, а с помощью атрибута `rows` указать число строк данного поля. Тег `<textarea>` парный.

2. `<select>` представляет собой выпадающий список, элементы которого указываются в тэге `<option>`. У этого тега существует несколько атрибутов. Чтобы сделать возможность выбора нескольких пунктов, необходимо задать атрибут `multiple`, в значении которого также необходимо указать `multiple`. В атрибуте `size` указывается число строк выпадающего списка, которые увидит пользователь в браузере. Если существует необходимость, чтобы какой-то из элементов выпадающего списка был выбран, то нужно этому элементу в тэге `<option>` задать атрибут `selected`. Тег `<select>` также парный.

3. Тег `<input />` . Он в зависимости от значения атрибута `type` будет выглядеть по-разному и иметь разные значения. Тег `<input />` — одиночный тег.

Рассмотрим более подробно тег `<input />`

- `<input type = "text" size = "30" placeholder = "Ваше Имя" >`
- `<input type = "password" size = "8" >`
- `<input type = "checkbox" checked = "checked" >`
- `<input type = "radio" name = "radio" >`
- `<input type = "file" >`
- `<input type = "submit" value = "Сохранить" >`
- `<input type = "reset" value = "Очистить" >`
- `<input type = "button" value = "просто кнопка" >`

У первого тега `<input />` значение атрибута `type= "text"`, это означает, что это обычное однострочное поле ввода. В значении атрибута `size` указывается размер этого поля в символах.

Атрибут `placeholder` выводит текст внутри текстового поля, который исчезает при начале ввода символов. `<input />` со значением `password` выглядит также, как и предыдущий, но он используется для ввода пароля и все символы будут показываться как точки, чтобы пароль нельзя было подсмотреть.

Следующие два варианта `<input type="checkbox" />` и `<input type="radio" />` предоставляют пользователю возможность право выбора некоторых пунктов. В первом случае есть возможность множественного выбора нескольких пунктов в виде флажков, а во втором — единственного выбора в виде переключателей. В обоих случаях есть возможность отметить любой пункт по умолчанию, если задать атрибут `checked="checked"`.

Кнопка выбора: `<input type="radio" name="pol">` - если необходимо выбрать что-то одно. Важно: для того чтобы кнопки работали в связке, у них должно быть одинаковое имя. В примере приведен пол, для кнопок М и Ж. Если на странице несколько групп таких кнопок, у них должны быть разные имена.

`<input type="file" />` предоставляет пользователю возможность выбора файла на своем компьютере, для того, чтобы загрузить его на сервер.

Оставшиеся 3 тега `<input />` — кнопки. `submit` отправляет данные на сервер. `reset` предназначена для того, чтобы очистить все поля в форме. А третья `type="button"` — просто кнопка, при нажатии на которой ничего не произойдет, но ее, например, можно привязать к событию `javascript`. У всех этих кнопок в значении атрибута `value` указывается тот текст, который будет отображен на кнопке.

Тег `<label>`

Тег `<label>` устанавливает связь между определенной меткой, в качестве которой обычно выступает текст, и элементом формы (`<input>`, `<select>`, `<textarea>`). Такая связь необходима, чтобы изменять значения элементов формы при нажатии курсором мыши на текст. Кроме того, с помощью `<label>` можно устанавливать горячие клавиши на клавиатуре и переходить на активный элемент подобно ссылкам.

Существует два способа связывания объекта и метки. Первый заключается в использовании идентификатора `id` внутри элемента формы и указании его имени в качестве атрибута `for` тега `<label>` . При втором способе элемент формы помещается внутрь контейнера `<label>` .

`<!-- Первый способ -->`

`<input id =" идентификатор ">`

`<label for =" идентификатор "> Текст </label>`

<!-- Второй способ -->

<label><input type = "..."> Текст </label>