

Министерство образования и науки Российской Федерации
Магнитогорский государственный технический университет
им. Г.И. Носова

М.В. Романова, Е.В. Чернова

РАЗРАБОТКА WEB-СТРАНИЦ И ПРЕЗЕНТАЦИЙ

Утверждено Редакционно-издательским советом университета
в качестве практикума

Магнитогорск
2017

УДК 37:004 (075)

Рецензенты:

Заведующий кафедрой экономики
ОУ ВО «Южно-Уральский институт управления и экономики»,
кандидат экономических наук, доцент
Л.В. Алферова

Директор по ИТ
ИП Полушкин Д.П.
Д.П. Полушкин

Романова, М.В.

Разработка Web-страниц и презентаций: практикум / М.В. Романова, Е.В. Чернова. – Магнитогорск: Изд-во Магнитогорск. гос. техн. ун-та им. Г.И. Носова, 2017. – 70 с.

Цель практикума – дать представление об основах Web-технологий: обучить навыкам работы в поисковых системах, научить создавать и опубликовывать в сети Интернет свои собственные статические и динамические Web-сайты.

Практикум предназначен для студентов непрофильных специальностей высших учебных заведений.

УДК 37:004 (075)

© Магнитогорский государственный
технический университет
им. Г.И. Носова, 2017
© Романова М.В.,
Чернова Е.В., 2017

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ	4
1. ОСНОВЫ ПОИСКА ИНФОРМАЦИИ В СЕТИ ИНТЕРНЕТ	5
1.1. Поисковые системы. Использование языка запросов	7
1.2. Практические задания по использованию поисковых систем.....	14
2. СОЗДАНИЕ WEB-ДОКУМЕНТОВ	15
Урок HTML № 1	15
Урок HTML № 2	23
Урок HTML № 3	30
Урок HTML № 4	36
Урок HTML № 5	49
Урок HTML № 6	56
3. ОСНОВЫ СОЗДАНИЯ ПРЕЗЕНТАЦИЙ	64
3.1. Практические задания по работе с PowerPoint	67
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	69

ВВЕДЕНИЕ

Цель данного практикума – предоставить студентам развернутый материал по основам:

- поиска информации в среде интернет;
- создания web-страниц на языке HTML;
- создания презентаций в Microsoft Power Point.

Практикум состоит из 3 частей. Каждая из этих частей направлена на изучение основ по одному из трех направлений. Каждая из трех частей практикума состоит из теоретической основы, примеров, пояснений и заданий для закрепления полученных знаний.

Настоящий практикум предназначен для студентов экономического и технических направлений, а также всем, кто рассматривает представленные в практикуме вопросы.

1. ОСНОВЫ ПОИСКА ИНФОРМАЦИИ В СЕТИ ИНТЕРНЕТ

Пользователь сети Интернет получает доступ к огромному числу информационных ресурсов. Например, количество имеющихся в Сети HTML-документов измеряется уже не десятками, а сотнями миллионов. В Интернет можно найти не только текст, но и программы, изображения, звуковые файлы и т.д. С одной стороны, в этом объеме информации, скорее всего, найдется та, которой вы интересуетесь, даже если сфера ваших интересов весьма специфична. С другой стороны, найти среди сотен миллионов Web-страниц именно те, которые представляют для вас интерес, – задача не простая.

Поиск необходимой информации в Интернете можно осуществлять различными способами:

- 1) поиск с помощью поисковых машин по ключевому слову;
- 2) поиск с помощью классификаторов поисковых машин;
- 3) каталоги и коллекции ссылок (более общие понятия);
- 4) рейтинги (самые популярные ресурсы);
- 5) форумы, конференции, чаты;
- 6) страницы ссылок («links») на тематических сайтах (редкие, специализированные вещи);
- 7) несетевые способы (советы друзей, знакомых; реклама в печатных изданиях).

В начале поиска информации необходимо определить ее тип. Условно можно выделить 4 типа информации:

- 1 тип — общая (например, «история Российской империи»);
- 2 тип — менее общая (например, «император Александр II»);
- 3 тип — конкретная (например, «реформы Александра II»);
- 4 тип — более конкретная (например, «отмена крепостного права»).

В зависимости от типа информации определяются и пути поиска.

1. Информация 1-го типа ищется с помощью классификаторов поисковых машин (например, Яндекс — www.yandex.ru). Если сразу сайты с требуемой информацией не находятся, то следует просматривать найденные по классификатору каталоги и страницы ссылок («links»), которые находятся на сайтах подобной тематике. Эти сайты приводятся в классификаторе по теме и найденных каталогах.

2. Информация 2-го типа ищется подобно поиску для 1-го типа, но с преимуществом поиска по каталогам и страницам ссылок.

3. Информация 3-го типа — по ключевым словам, которые вводятся в строку поиска поисковых машин, каталогам, страницам ссылок

4. Информация 4-го типа — по подробным данным, которые вводятся в строку поиска. Данные находятся согласно способам поиска изложенных для 2-го и 3-го типов.

Примеры:

Поиск по 1-му типу. Требуемая информация: «История Российской империи».

Заходим на сайт поисковой системы Яндекс, переходим в «Каталог»: Наука и образование -> Общественные науки -> История. По описанию темы находим сайт <http://rus-hist.on.ufanet.ru>. Если в нем нет необходимой информации, то переходим на страницу ссылок этого сайта. На ней имеются ссылки на каталоги ресурсов: www.history.ru, <http://www.lants.telhir.ru/history/index.htm>. В них, скорее всего, будут найдены сайты по заданной теме.

Поиск по 2-му типу. Требуемая информация: «Император Александр II».

Поиск осуществляется аналогично предыдущему, но больше внимания уделяется работе с каталогами www.history.ru.

<http://www.lants.telhir.ru/history/index.htm>.

Поиск по 3-му типу. Требуемая информация: «Реформы Александра II».

Здесь появляется новый способ поиска – по ключевым словам. Пишем в строке поиска Яндекса «Реформы Александра II». Результат для просмотра – 1790 страниц, которые находятся на 170 сайтах, в число которых входят и каталоги. Для сужения информации можно добавить новые ключевые слова – дополнительные факты в уже найденном подборе сайтов, например: «1860–1870 гг.» и т.д. В других поисковых машинах набирается полностью «Реформы Александра II в 1860–1870 гг.». Для поиска заданной информации можно еще использовать «links», которые приводятся на найденных сайтах

Поиск по 4-му типу. Требуемая информация: «Отмена крепостного права в России».

Ищем по ключевым словам в Яндексе: «отмена крепостного права». Выдается результат – 4077 страниц. Для сужения информации в поиск по ключевым словам вводим «1861» – год подписания манифеста. Результат – 1541 страниц для просмотра. Далее вводим «Александр II» получается 591 страница, но для лучшего результата добавляем в найденном материале слово «манифест» появляется результат — 368 страница. В этом перечне можно уже искать нужную информацию. При необходимости можно продолжить уменьшение количества страниц для просмотра, вводя новые ключевые слова. Следует отметить, что все дополнительные слова найдены в биографии Александра II по методике, изложенной для 2-го типа поиска.

1.1. Поисковые системы. Использование языка запросов

Самый простой, распространенный и, в тоже время, наименее эффективный способ поиска – набор фразы, в том виде, как она должна примерно выглядеть или набора слов, которые, по Вашему мнению, должны присутствовать в подобных работах. Статистика показывает, что успешность подобного рода поиска составляет в среднем для гуманитарных элементов около 10%. В тоже время, если абсолютно точно известна техническая марка, эффективность даже такого простого варианта поиска может достигать почти 70%. Отсюда уже сразу напрашивается вывод, что и подобным простейшими вариантами не стоит пренебрегать, особенно если речь идёт о поиске известных технических элементов марок, конструкций.

Поиск в компьютерных сетях становится искусством и требует вполне определённых знаний, которыми современному человеку необходимо овладеть.

Основные поисковые машины

В настоящее время в отечественных ресурсах действует довольно много поисковых машин. Есть фирменные, и даже индивидуальные сайты, которые включают те или иные поисковые машины. Поисковые элементы существуют практически на всех Интернет-газетах и учебных порталах. Наиболее распространённые из поисковых систем: yandex.ru, google.ru, rambler.ru, yahoo.ru, km.ru, top100. Этим их список далеко не ограничивается.

Причины, по которым документ не может быть обнаружен

Прежде всего, требуемое не будет обнаружено поисковой машиной, если на ней данные ресурсы не прописаны.

Запомните! Недостаточно разместить, что-либо в Интернет. Необходимо прописать это на соответствующих поисковых машинах. Можно расположить материал в Интернет, зная его адрес. Однако, пока он не «прописан» на одной из поисковых машин, найти его стороннему человеку будет очень сложно, конечно если ему не дан адрес или DNS. Через некоторое время и этот сайт будет обнаружен так называемыми «поисковыми роботами» или «пауками-поисковиками» и, только тогда, возможно, он начнёт посещаться.

Впрочем, есть возможность программно запретить регистрацию всего сайта, или отдельных элементов его, как всеми, так и определёнными поисковыми системами. Для подобных целей служат так называемые мета-теги или мета-определители. Следующее – это специализация поисковых машин. Так top100 относится к поисковым машинам специализированного типа и больше предназначена для поиска технических элементов компьютеров и справок по ним. Поисковые машины yandex.ru, google.ru, rambler.ru, yahoo.ru, km.ru относятся к группе общего назначе-

ния. Именно с них следует начинать поиск в Интернете. Причем, поисковая система «Кирилл и Мефодий» на km.ru все же чаще используется для розыска материалов в разного рода энциклопедиях и учебных пособиях, словарях, которые вышли после 1990 г. На AlfaVista – наиболее полно представлены материалы, связанные с коммерческой деятельностью, Google – более специализируется на оперативных новостях. На машинах Fast, Инфоарт, ИППИ РАН, Русский интернет и Апорт часто можно бывает найти информацию, связанную с куплей-продажей.

Следует учитывать, что полностью адрес поисковых машин обычно включает префикс и выглядит примерно так: <http://www.ya.ru>. Некоторые поисковые системы используют чужие поисковые системы. Так система поиска mail.ru на сегодняшний день использует поисковую машину Rambler.ru, но выводит информацию в собственном интерфейсе.

Задача поиска

Перед тем, как начать поиск, прежде всего, следует четко сформулировать задачу. До тех пор, пока этот момент не отработан об успешности, а тем более оптимизации поиска, не может быть и речи. Объем информации в Интернете слишком велик, и как нигде здесь верна поговорка «Время – деньги». Поэтому в нём нельзя искать по принципу «Пойди туда, не знаю куда. Возьми то, не знаю что». Иначе результат будет плачевный.

Язык поисковых запросов

Поисковый запрос может состоять из одного или нескольких слов, в нем могут присутствовать различные знаки препинания. Составлять самые простые запросы можно и, не вдаваясь в тонкости языка запросов. Так, если ввести в поисковую строку несколько слов без знаков препинания и логических операторов, будут найдены документы, содержащие все эти слова, да ещё и если они находятся на строго ограниченном расстоянии друг от друга. Такой поиск с довольно большой вероятностью закончится отрицательным результатом. И здесь о понятии успешности не может быть и речи. Знание и правильное применение языка запросов поисковой машины поможет сделать Ваш поиск быстрым и эффективным.

Регистр

В общем случае, регистр написания поисковых слов и операторов значения не имеет, то есть слова «дом», «Дом», «дОМ», «доМ» и «ДОМ» – все будут восприниматься одинаково. Это полностью касается и латиницы. Так, «Not» и «nOt», и даже «noT», «not» и «NOT» все они для поиска однообразны. И лишь иногда, в целях повышения качества поиска, регистр поискового запроса принимается во внимание.

Так, если запрос состоит из двух, трех или четырех слов, каждое из которых написано с большой буквы, то предполагается поиск по имени собственному, и автоматически производится изменение ограничения расстояния между словами запроса со значения по умолчанию на вели-

чину $(n-1) \cdot 2$, где n – количество слов запроса. Это позволяет находить группу слов запроса, внутри которой есть не более одного «лишнего» слова или знака препинания, например: «М. Лермонтов», «Баден-Баден», «Шарль де Костер», «Федор Михайлович Достоевский» и т.п.

Операторы

Современные поисковые машины используют операторы для поиска в компьютерных сетях. Под оператором здесь понимается инструкция поиска. По этой причине запрос, включающий несколько слов, может содержать операторы. Поиск указанных операторов в документе не производится. Они служат лишь инструкцией поисковой машине. По этой причине они имеют свою орфографию и синтаксис. Все операторы поисковой машины бинарные связи, то есть имеют левую и правую часть, каждая из которых также является запросом (по умолчанию состоящим из одного слова). Для изменения сферы действия операторов (например, группировки нескольких слов запроса в аргумент оператора) применяются скобки и кавычки.

Два запроса, соединенные оператором AND (логическое И) образуют сложный запрос, которому удовлетворяют только те документы, которые одновременно удовлетворяют обоим этим запросам. Иными словами, по запросу «самолеты AND авиация» найдутся только те документы, которые содержат и слово «самолеты», и слово «авиация» одновременно. Если эти слова встретятся в текстах отдельно, поисковая машина Вам эти тексты не покажет или покажет в конце списка найденных материалов, как только частично отвечающие запросу. Контрольная проверка сформулированного запроса может быть осуществлена по логике фразы: «И ТО И ДРУГОЕ».

Сложному запросу, состоящему из двух запросов, соединенных оператором OR (логическое ИЛИ) соответствуют все документы, удовлетворяющие хотя бы одному из этих двух запросов. По запросу «самолёты OR авиация» будут найдены документы, в которых присутствует хотя бы одно из этих двух слов, либо оба эти слова вместе. Логическая контрольная проверка может быть осуществлена по фразе: «ИЛИ ТО ИЛИ ДРУГОЕ, ИЛИ ОБА ВМЕСТЕ».

Оператор NOT (логическое И-НЕ) образует запрос, которому отвечают документы, удовлетворяющие левой части запроса и не удовлетворяющие правой. Так, результатом поиска по запросу «самолёты NOT авиация» будут показаны все документы, в которых есть слово «самолёты» и при этом в данном документе на установленном расстоянии отсутствует слово «авиация». Контрольная логическая проверка может быть осуществлена по фразе: «ПЕРВОЕ, НО ТОЛЬКО БЕЗ ВТОРОГО».

Обратите внимание! Если оператор в явном виде не указан, используется оператор по умолчанию AND. Как результат: будут выведены документы, содержащие все слова запроса. Так, запрос информация шко-

ла успеваемость будет истолкован как информация AND школа AND успеваемость. На странице Расширенного поиска оператор по умолчанию можно заменить на OR (искать слова запроса, и выводить даже если встречается хотя бы одно из них).

Каждый из операторов имеет своё сокращенное обозначение, соответствие их приведено в табл. 1.

Таблица 1

Операторы и их сокращенное обозначение

Оператор	Приоритет	Сокращенное обозначение
AND	Высший	&
NOT	Нижний	!
OR	Высший	

Приоритеты логических операторов. Запрос из нескольких слов, перемежающихся операторами, будет истолкован в соответствии с их приоритетом. Операторы AND и NOT традиционно имеют более высокий приоритет (таблица 1). В этой связи запрос из нескольких слов при обработке сначала группируется по операторам AND и NOT, и лишь потом по операторам OR. (Вспомните, для примера, приоритеты арифметических действий. Так знаки «*» и «/» имеют равный приоритет, но больший чем знаки действий «+» и «-»). Изменить порядок группировки можно, как и в математике, используя скобки.

Скобки

При построении запросов иногда возникает необходимость объединения слов запроса в группы, которые будут аргументами некоторого оператора. Такие группы заключаются в скобки. Как правило, в большинстве поисковых машин используют только скобки круглого вида: ().

Часть запроса, заключенная в скобки, сама является запросом, и на нее распространяются все правила языка построения запросов. Использование скобок позволяет строить вложенные запросы и передавать их операторам в качестве аргументов, а также перекрывать приоритеты операторов, принятые по умолчанию.

Если запрос был задан без скобок, например: «машина самолет | аэродром», он эквивалентен запросу «машина AND самолет OR аэродром». В соответствии с приоритетами операторов, означает «найти документы, содержащие либо слова «машина» и «самолет» в одном документе, либо найти в нём слово «аэродром». В то время как запрос со скобками «машина (самолет | аэродром)» равносильен запросу «машина AND (самолет OR аэродром)», что означает «найти документы, содержащие слово машина и одно из слов самолет или аэродром».

Кавычки

Для поиска цитат можно использовать типовые (двойные) кавычки, вида « или ». Слова запроса, заключенного в двойные кавычки, ищутся в документах именно в том виде и в том порядке, как они были заданы Вами в запросе. Это весьма эффективный способ поиска. Его применение обеспечивает успешность более 60%. Здесь важно учитывать только два момента:

первое – цитирование должно быть точным; поскольку уже сама форма запроса требует искать «как есть», т.е. без изменения по падежам, числам и лицам;

второе – материал, который Вы ищите, как минимум должен быть в Интернете.

Запомните, если при такой форме поиска будет допущена грамматическая ошибка, поиск даст отрицательный результат.

Таким образом, двойные кавычки можно использовать и просто для поиска слова в заданной форме (по умолчанию слова находятся во всех формах). Например, запросу: теплоход «причалил» посадка удовлетворяет документ, содержащий текст «...теплоход причалил для осуществления посадки пассажиров...», и не удовлетворяют документы: с текстом «...теплоход подошел к причалу для посадки пассажиров...», или «...теплоход был причален и на него была проведена посадка пассажиров...» и т.п.

Метасимволы или маски

Далеко не всякая поисковая машина может поддерживать поиск строк с использованием метасимволов {«*», «?»}, которые обычно используются в значении «любая подстрока» и «произвольный (любой) одиночный символ» соответственно. Тем не менее, эти операторы нередко бывают зарезервированы для подобного использования в будущем. Реализация подобного есть, например, на www.mail.ru.

Применение языка запросов

Каждый запрос, адресованный поисковой машине, обрабатывается в соответствии с правилами языка запросов. Некоторые слова и символы трактуются как операторы языка запросов и обрабатываются специальным образом. *Фактически, языком запросов описывается некая формула, которая используется при поиске – каждый из документов «сопоставляется» с ней, и результатом поиска являются только те документы, которые ей удовлетворяют.*

Например, запросу самолет удовлетворяют все документы, в которых хотя бы раз встретилось слово «самолет» в любом падеже, включая как единственное, так и множественное число: *самолёт, самолеты, самолету, самолетом, самолете, самолетах, самолётам* и т.п. Запросу, состоящему из нескольких слов, удовлетворяют документы, содержащие каждое из этих слов в любой форме (при некоторых условиях). Вопрос

соответствия документа более сложному запросу определяется логикой операторов и конструкций запроса.

При этом в поисковую строку Вами вводятся, как считают ключевые слова. Строго говоря – это не совсем так. Следует отличать поиск действительно по ключевым словам, указанным как таковые на данном документе в Интернет (вариант используется в расширенном поиске). Это делается в помощью специальных мета-определителей в головной части описания гипертекстового документа. Согласитесь, то, что Вы считаете, ключевым словом совсем не обязательно должно быть описано автором работы как таковое. В этой связи правильное будет назвать подобное – словами, которые, по Вашему мнению, обязательно должны быть в требуемых материалах.

Морфология языка запросов

По каждому слову запроса поиск ведется с учетом правил словоизменения соответствующего языка. Поисковая машина «понимает» и «различает» слова русского и английского языков — по умолчанию. Поиск ведется с учетом изменения по склонению, спряжению, числу и лицу, т.е. по всем формам слова; включая даже совпадающие по смысловому значению.

Например, при поиске по слову «человек» будут также найдены документы, содержащие слова «о человеке», «человеку», «человеком», 'человека' и т.п., но и, что особенно важно, даже слова «люди», «людям», 'людей' окажутся соответствующими запросу. Чтобы провести поиск только по одной строго определенной форме слова, нужно взять его в двойные кавычки или воспользоваться поиском точной фразы в расширенном поиске.

Стоп-слова

Некоторые слова и символы по умолчанию исключаются из запроса в связи с их малой информативностью. Это так называемые стоп-слова – самые частотные слова русского и английского языков, например, предлоги, частицы и артикли. Очень часто опускаются апострофы и тире. Большинство поисковых машин их знают и сами освобождают от них сформулированный Вами запрос. Это связано с тем, что факт присутствия этих слов может заметно замедлить поиск и отрицательно повлиять на результат и полноту, а, в конечном итоге, и успешность. В то же время есть возможность обозначить необходимость этих слов в запросе. Для этого как мы уже знаем, следует взять запрос в двойные кавычки или воспользовавшись поиском точной фразы в расширенном поиске.

Ограничение расстояния

Если запрос составлен из нескольких слов без применения операторов и конструкций языка запросов, то машина будет пытаться найти документы, в которых встречаются все слова запроса. При этом для каждого запроса всегда вводится так называемое ограничение контекста –

положительное число. По умолчанию принимается расстояние равное 40 словам. Документ, в котором встретились все слова запроса, будет выдан только в том случае, если расстояние между словами запроса будет меньше этого числа. Например, по запросу российская армия будут найдены те документы, в которых слова российская и армия хотя бы один раз встретятся на расстоянии менее чем в 40 слов друг от друга. Причем, чем ближе они друг к другу окажутся – тем выше будет и уровень их соответствия запросу.

Значение ограничения контекста можно изменять конструкцией «(число, запрос)», где число – любое положительное число, запрос – любой корректный с точки зрения поисковой машины, запрос, состоящий более чем из одного слова (очевидно, ограничение расстояния между словами в случае однословного запроса не имеет смысла). Таким образом, по запросу (2, армии НАТО) проводится поиск только тех документов, в которых между словами НАТО и армии хотя бы раз не стоит ни одного слова (поскольку лишь в случае их непосредственного соседства разница в порядковых номерах слов меньше 2, т.е. равна 1).

Ненайденные слова

Если запрос состоит из нескольких слов, и при этом подобный набор слов вообще не удалось найти в Интернете, будут выданы результаты поиска по частичному соответствию данному запросу, из которого отсутствующие в Интернет слова исключены.

При этом на странице результатов поиска будет выведена соответствующая диагностика. В большинстве случаев она выдается в процентах соответствия (релевантности), которые рассчитываются по специальной формуле с учетом полноты соответствия каждого из слов, расстояния между словами запроса, соответствие лица и числа, окончания и т.п.

Базовый алгоритм поиска

1. Прежде всего, определите, к какому разделу относится искомая информация.
2. Определитесь, какую именно поисковую машину Вам лучше использовать. Учитывайте её специфику.
3. Выявите основные слова (или словосочетания), которые характерны для той информации, которую Вы ищете. Попробуйте выявить ключевые слова.
4. Задавая отдельные слова и фразы, учитывайте язык запросов, операторы поиска, логические связи искоемых слов и т.п. элементы, которые делают поиск значительно более эффективным.
5. Если Вам известна точная фраза из искоемых материалов, используйте её как цитату.
6. После проведения предварительного поиска (или первой ступени поиска), если выдано довольно большое количество документов, Вы можете воспользоваться элементом уточнения: "Искать в избранном".

При этом задание поиска следует обязательно уточнить, введя хотя бы одно новое слово или ограничение, которое по Вашему мнению поможет провести правильный отбор требуемого материала.

7. Грамотно располагайте материал и проводите его сортировку по удобным для Вас признакам. Это заметно ускорит последующую обработку материала.

8. Фиксируйте все интересные для Вас найденные адреса документов в Интернете (например, с помощью программы «Блокнот»). Затем выберите среди них нужные для выполнения конкретной работы.

1.2. Практические задания по использованию поисковых систем

Задание 1. Информационный поиск

1. Используя шаблон простого поиска в поисковой системе, найдите документы содержащие полное вхождение поискового слова – **ресурс**.

2. Используя шаблон простого поиска в поисковой системе, найдите документы по вхождению начала слова (корня) – **информация**. Приведите примеры найденных словоформ.

3. Используйте возможность искать в результатах поиска для нахождения документов по запросу – **информационный ресурс**.

4. Найдите документы, содержащие все слова из запроса – **мировые информационные ресурсы**.

5. Найдите документы, содержащие различные формы хотя бы одного поискового слова из запроса – **поисковые систем сети интернет**.

6. Используя логические связи &, | и группирование слов (), подготовьте запрос для нахождения документов, содержащих слово 'машина' и одно из слов 'самолет' или 'аэродром'.

7. Используя логические связи, группирование слов или ограничивая дату поиска в расширенной форме выясните текущие цены на баррель нефти.

8. Используя расширенный поиск, найдите картинки с названием «cat».

Задание 2. Использование страниц с поиском по нескольким поисковым системам

1. Осуществите запрос, состоящий из одного слова с использованием поисковой системы Rambler (www.rambler.ru) и метапоисков:

Yandex-Rambler – (<http://www.comptek.ru/ramb.html>),

Yandex-AltaVista – (<http://www.comptek.ru/alta.html>).

2. Проанализируйте статистику результатов, выдаваемую в каждом случае.

Задание 3. Сравнение качества поиска поисковых систем

Поисковая машина характеризуется двумя важнейшими параметрами: точностью и полнотой (полнота есть отношение количества найденных релевантных документов к полному количеству релевантных документов в базе данных).

Пример. Пусть по запросу найдено 50 документов. После просмотра их всех пользователь принимает решение, что 30 документов релевантны запросу, а 20 нерелевантны. Сплошной просмотр всей базы данных показывает, что в ней содержится 100 документов, релевантных запросу. Отсюда получаем, что полнота $30/100 = 0,3$; точность $30/50 = 0,6$. Как правило, улучшая один из названных параметров, ухудшаешь другой.

Просмотрите несколько поисковых систем (не менее 6) задавая запрос по одноименной теме и проведите оценку качества поиска в данных системах. Результатом работы будет являться заполненная таблица:

Название поисковой системы	Полнота поиска	Точность поиска	Быстродействие поиска	Удобство представления результатов

Примечание. Параметр «быстродействие поиска» – это интервал времени между моментом формулировки запроса и получением ответа на него); «удобство представления результатов» оценивается по 3-балльной шкале.

2. СОЗДАНИЕ WEB-ДОКУМЕНТОВ

Урок HTML № 1

Язык HTML достаточно прост в изучении, интуитивен и близок к разговорному английскому языку.

Язык HTML состоит из набора тэгов.

Тэг – элемент HTML, представляет из себя текст, заключенный в угловые скобки $\langle \rangle$, является активным элементом, изменяющим представление следующей за ним информации. Может иметь некоторые атрибуты. Обычно имеются два тэга - открывающий и закрывающий.

В языке HTML существуют 3 основных тэга, которые присутствуют на всех страницах, вне зависимости от их содержания:

- **<HTML>** – Сообщает браузеру, что документ создан на языке HTML.
- **<HEAD>** – Содержит вводную и заголовочную части HTML-документа.

- **<BODY>** – Содержит основной текст и информацию страницы.

Эти тэги необходимы Web-браузеру для определения различных частей HTML-документа, они не оказывают прямого влияния на внешний вид WEB-страницы. Они необходимы для того, чтобы последующие нововведения в HTML правильно интерпретировали вашу страницу, а также для того, чтобы она выглядела одинаково в часто используемых браузерах (IE, Mozilla, Opera и др.).

Эти тэги сообщают браузеру, что текст между ними следует интерпретировать как HTML-текст. Поскольку документы HTML только текстовые, тэг **<HTML>** говорит о том, что файл написан на языке HTML (Hyper Text Markup Language - Язык гипертекстовой разметки).

Создавая новый HTML-файл, в первую очередь необходимо ввести данную пару тэгов. Для этого наберите **<HTML>** в самом начале текста. Затем наберите его напарника - **</HTML>** - в конце. Теперь весь текст, написанный между ними, будет принят браузерами за текст HTML. Тэг с символом **</>** является закрывающим тэгом. Большинство HTML-тэгов парные: один открывает (**<HTML>**), другой закрывает (**</HTML>**). Их действие распространяется только на тот текст, который находится между ними.

Итак, сейчас наша страница выглядит таким образом:

файл: index.html
<HTML> </HTML>

Открыв в браузере эту страницу, мы ничего не увидим, пока мы не завершили работу над **<HEAD>** и **</HEAD>**.

Теперь введем тэги **<HEAD>** и **</HEAD>**. Они должны быть между тэгами **<HTML>** и **</HTML>**. Эти тэги отмечают ту информацию в нашем документе, которая будет служить названием.

В этих тэгах должна содержаться следующая информация:

- **<TITLE>Моя страница</TITLE>** – Между этими двумя тэгами необходимо ввести заголовок Вашей странички.

- **<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251">** – Эту строчку нужно вставить, если Вы создаете свой документ на блокноте или другом простейшем текстовом редакторе, которая будет сообщать браузерам кодировку языка, на которой написана страница.

Так же не забываем указать тэги **<BODY></BODY>**, которые расположены, как и тэги **<HEAD>**, внутри тэга **<HTML>**, сразу за закрывающим тэгом **</HEAD>**.

Посмотрим, как выглядит страничка теперь:

файл: index.html

```
<HTML>
<HEAD>
  <TITLE>Design LAB</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; char-
set=windows-1251">
</HEAD>
<BODY>
</BODY>
</HTML>
```

Созданная нами страница до сих пор является пустой, в ней появилось только название самой страницы. Также следует отметить, что тэги создаются не друг за другом, а в виде нисходящей иерархии, что облегчает читаемость кода. Далее перейдем к изменению страницы.

Тэги `<BODY>` и `</BODY>`. С их помощью мы сможем изменить стиль страницы, перед тем как набрать текст, и наполнить страницу содержимым. К примеру, нам необходимо сделать так, чтобы текст был белым на черном фоне, а ссылки (посещенные, не посещенные или локальные) – светло-зелеными. В этом случае, необходимо дополнить тэг `<BODY>` следующими параметрами:

файл: index.html

```
<HTML>
...
<BODY BGCOLOR="BLACK" TEXT="WHITE" LINK="LIME"
ALINK="LIME" VLINK="LIME">
...
</BODY>
</HTML>
```

- **BGCOLOR="BLACK"** – цвет фона – черный.
- **TEXT="WHITE"** – цвет текста – белый.
- **LINK="LIME"** – цвет не посещенной ссылки – салатовый.
- **VLINK="LIME"** – цвет посещенной ссылки – салатовый.
- **ALINK="LIME"** – цвет локальной ссылки – салатовый.

Итак, теперь мы, наконец, попробуем создать текст. К примеру, нам нужно сделать ссылку на страничку с информацией о создателе сайта.

файл: index.html

```
<HTML>
<HEAD>
  <TITLE>Design LAB</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; char-
set=windows-1251">
</HEAD>
<BODY BGCOLOR="BLACK" TEXT="WHITE" LINK="LIME"
ALINK="LIME" VLINK="LIME">
  На этой страничке есть информация про
меня
</BODY>
</HTML>
```

Открыв эту страничку в браузере в таком виде, мы получим примерно следующее.

preview: index.html

На этой страничке есть информация **про меня**

При наведении курсора мыши на надпись **про меня** он (курсор) примет форму руки. И далее, кликнув по надписи, мы попадем на страничку **about.html**, где расположена информация об авторе, если данная страница не создана, то браузер нам сообщит, что он не может найти запрашиваемую страницу.

У многих браузеров существует история посещенных страниц. Скажем, если мы когда-то были на нашем сайте, и заходили в раздел **про меня**, то мы увидим эту ссылку другим цветом. По умолчанию этот цвет фиолетовый, но изменив в `<BODY>` значение `VLINK="LIME"`, мы увидим посещенную ссылку салатовым цветом. А еще можно сделать так, чтобы кликая на ссылку, скрывающаяся за ней страница открывалась в новом окне. Синтаксис таков:

```
<a target="_blank" href="about.html">Про меня</a>
```

Отсюда видно, что кроме параметра **HREF** (что значит «направление», «расположение»), здесь присутствует `target="_blank"`. Этот параметр и дает команду открывать ссылку в новом окне.

Цветовая гамма в HTML

Язык HTML понимает 2 вида цветовых переменных: HEX и Color. **Color** – это ввод цвета словом, как в английском языке (red – красный, blue – синий и т.д.).

Палитра текстовых цветов:

 red	 cyan	 lime	 pink	 fuchsia
 green	 gray	 maroon	 navy	 olive
 blue	 purple	 silver	 teal	 yellow

HEX-параметр задается в виде 6-значного числа в 16-ой форме. Давайте посмотрим на эту таблицу:

 #110000	 #330000	 #550000	 #990000	 #ff0000
 #001100	 #003300	 #005500	 #009900	 #00ff00
 #000011	 #000033	 #000055	 #000099	 #0000ff

В **HEX** палитре числа представлены парами, т. е. первая пара влияет на красный цвет, вторая пара чисел влияет на зеленый цвет, и третья пара – на синий (**#RRGGBB**).

Заголовки

Подобно названию страницы, заголовки должны быть краткими и полезными. Существует шесть размеров шрифта заголовков (они пронумерованы от одного до шести, причем первый номер соответствует самому крупному размеру шрифта).

Пример заголовка - размер 1
Пример заголовка - размер 2
Пример заголовка - размер 3
Пример заголовка - размер 4
Пример заголовка - размер 5
Пример заголовка - размер 6

Заголовки отображаются Web-браузерами намного крупнее и жирнее стандартного текста. Это хороший способ выделять различные части Web-страницы.

Чтобы воспользоваться заголовком с размером букв первого номера, нужно обозначить выделяемый текст тэгами **<H1>** и **</H1>**. Как уже отмечалось выше вводимый нами текст должен находиться между тэгами **<BODY>** и **</BODY>**.

файл: index.html

```
<HTML>
<HEAD>
  <TITLE>Design LAB</TITLE>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html;
charset=windows-1251">
</HEAD>
<BODY BGCOLOR="BLACK" TEXT="WHITE" LINK="LIME"
ALINK="LIME" VLINK="LIME">
  <H1>Моя Домашняя Страничка</H1>
  На этой страничке есть информация <a href="about.html">про ме-
ня</a>
</BODY>
</HTML>
```

И вот что мы получим в результате:

preview: index.html

Моя Домашняя Страничка

На этой страничке есть информация **про меня**

Написав **<H1>**, мы активизировали команду "Заголовок #1". Написав **</H1>**, мы закрыли эту команду. Весь текст, который находился между этими тэгами, был подвластен команде "Заголовок #1".

Набор текста

Процедура набора текста в HTML-документ является самой простой, вы просто пишете весь необходимый вам текст между тэгами **<BODY>**.

Пример ввода текстовой информации:

Здравствуйте. Меня зовут Петров Илья. Мои любимые сайты:
Desing Laboratory (<http://ertu.wsdu.pl/~Sam/dlab>)
Bjork (<http://bjork.by.ru>)
Prodigy (<http://prodiga.by.ru>)
iPoizon.com (<http://iPoizon.com>)

Но запустив нашу страничку на браузере, мы увидим:

preview: index.html
Моя Домашняя Страничка Здравствуйте. Меня зовут Петров Илья. Мои любимые сайты: Desing Laboratory (http://ertu.wsdu.pl/~Sam/dlab) Bjork (http://bjurk.by.ru) Prodigy (http://prodiga.by.ru) iPoizon.com (http://iPoizon.com)

Все смешалось. Необходимо отформатировать текст. Существует тэг **
**. Этот тэг нужно вставлять в месте, где нужно делать перенос строки, процедура аналогичная нажатию клавиши «enter» в текстовом редакторе.

После обработки текста тэгом **
** результат выглядит следующим образом:

preview: index.html	файл: index.html
Моя Домашняя Страничка Здравствуйте. Меня зовут Петров Илья. Мои любимые сайты: - Desing Laboratory (http://ertu.wsdu.pl/~Sam/dlab) - Bjork (http://bjurk.by.ru) - Prodigy (http://prodiga.by.ru) - iPoizon.com (http://iPoizon.com)	<pre> <HTML> <HEAD> <TITLE>Design LAB</TITLE> <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=windows-1251"> </HEAD> <BODY BGCOLOR="BLACK" TEXT="WHITE" LINK="LIME" ALINK="LIME" VLINK="LIME"> <H1>Моя Домашняя Страничка</H1> Здравствуйте. Меня зовут Петров Илья. Мои любимые сайты:
 - Desing Laboratory (http://ertu.wsdu.pl/~Sam/dlab)
 - Bjork (http://bjurk.by.ru)
 - Prodigy (http://prodiga.by.ru)
 - iPoizon.com (http://iPoizon.com)
 </BODY> </HTML> </pre>

Существует также тэг параграфа, т.е. небольшой отступ от начала строки, аналогичный нажатию клавиши «TAB» в текстовом редакторе. Это тэг **<dd>**. Заменяем знаки «-» на **<dd>**:

preview: index.html	файл: index.html
Моя Домашняя Страничка Здравствуйте. Меня зовут Петров Илья. Мои любимые сайты: Desing Laboratory (http://ertu.wsdu.pl/~Sam/dlab) Bjork (http://bjurk.by.ru) Prodigy (http://prodiga.by.ru) iPoizon.com (http://iPoizon.com)	<pre> <HTML> ... Здравствуйте. Меня зовут Пет- ров Илья. Мои любимые сай- ты:
 <dd>Desing Laboratory (http://ertu.wsdu.pl/~Sam/dlab)
 <dd>Bjork (http://bjurk.by.ru)
 <dd>Prodigy (http://prodiga.by.ru)
 <dd>iPoizon.com (http://iPoizon.com)
 ... </HTML> </pre>

Помните, что этот тэг не работает в ранних версиях различных браузеров.

Разделители

Хотя простые линии очень полезны, они довольно скоро надоели создателям Web-страниц. Поэтому тэг **<hr>** был доработан до его теперешнего, "резинового" в обращении, вида. Теперь помимо тонкой линии поперек страницы вы можете применить и другой тип линии, используя расширения тэга **<hr>**.

Простая линия <hr>
Линия без тени <hr noshade>
Линия размером в 10px <hr size="10">
Линия шириной в 80% <hr width="80%">
Оранжевая линия <hr color="orange">
А теперь сгармонируем все параметры в одну линию <hr noshade width="80%" color="orange" size="10" >

Параметры разделительных линий:

- `<hr noshade >` – По умолчанию, линия отбрасывает трудно-видимую тень. Для того чтобы тени от линии не было, необходимо использовать такой синтаксис.

- `<hr size="10">` – По умолчанию, линия имеет размер 1 пиксель. Но мы можем его менять. Для этого необходимо вписать `size="ваше число"`, где **ваше число** - число пикселей, в нашем случае 10.

- `<hr width="80%" >` – По умолчанию, линия имеет ширину равную 100%, т.е. во всю длину окна браузера. В данном случае мы изменили этот параметр на 80%.

- `<hr color="orange" >` – По умолчанию, линия имеет прозрачный цвет. Но из-за тени, нам кажется, что линия серая. Мы можем изменить цвет линии. Синтаксис виден из примера.

Все эти параметры могут сочетаться и использоваться параллельно.

Задание к 1 лабораторной работе:

1. Создать свою страницу в блокноте (index.html).
2. Разместить на странице информацию о себе.
3. Отформатировать текст на странице при помощи тэгов `<dd>`, `<br>`, `<hr>`.

Урок HTML № 2

Данная практическая работа рассматривает варианты стилового оформления текстовой информации страницы.

Выравнивание текста по центру

Выравнивание текста по центру позволяют сделать тэги `<center>` и `</center>`. При использовании этих тэгов заголовки или текст не будут прижаты к левому краю, они равномерно заполнят область вывода текста. Чтобы выровнять текст по центру, просто вставьте тэги `<center>` и `</center>` в начале и в конце нужного фрагмента:

```
<center><h1>Пример заголовка - размер 6</h1></center>
```

Вот что мы получим в результате:

Пример заголовка - размер 6

Вы можете центрировать заголовки, текст, абзацы и веб-графику.

Полужирный шрифт

Тэги `<b>` и `</b>` позволяют отображать различные слова и фразы вашего документа полужирным шрифтом. Текст между этими тэгами выглядит толще обычного текста, чем приятно выделяется. Чтобы разметить текст как полужирный, вставьте `<b>` и `</b>` в начале и в конце текста:

полужирный текст выделяется на фоне обычного

Вот что мы получим в результате:

полужирный текст выделяется на фоне обычного

Курсив

Тэги `<i>` и `</i>` позволяют отображать различные слова и фразы вашего документа курсивным шрифтом. Чтобы разметить текст как курсивный, вставьте `<i>` и `</i>` в начале и в конце текста:

<i>курсивный</i> текст выделяется на фоне обычного

Вот что мы получим в результате:

курсивный текст выделяется на фоне обычного

Мерцание

Тэг мерцания `<blink>` `</blink>` не работает в браузерах Microsoft Internet Explorer. Вообще этот тэг самый раздражающий из всех существующих тэгов в HTML, но разумное использование тэгов `<blink>` и `</blink>` может вполне приукрасить вашу страничку.

<blink>мерцающий</blink> текст выделяется на фоне обычного

Вот что мы получим в результате:

мерцающий текст выделяется на фоне обычного

Подчеркивание

Существует в HTML также тэг, который используется для подчеркивания какого-либо текста. Это тэги `<u>` и `</u>`.

<u>подчеркнутый</u> текст выделяется на фоне обычного

Вот что мы получим в результате:

подчеркнутый текст выделяется на фоне обычного

Перечеркивание

Существует в HTML также тэг, который используется для перечеркивания какого-либо текста. Это тэги `<s>` и `</s>`.

<u>перечеркнутый</u> текст выделяется на фоне обычного

Вот что мы получим в результате:

~~перечеркнутый~~ текст выделяется на фоне обычного

Размеры текста

Иногда при написании текста в веб-странице необходимо слегка выделить определенные фрагменты текста, чтобы читатель обратил на них особое внимание. HTML предлагает ряд возможностей для такого выделения. Тэги `<big>` и `<small>` один из таких способов. Они не оказывают существенного влияния на вид текста, но вызывают некоторые изменения в размере шрифта.

Шрифт текста между тэгами `<big>` и `</big>` немного крупнее обычного текста. Как вы догадаетесь, действие тэгов `<small>` и `</small>` прямо противоположно. Шрифт между ними становится немного меньше.

`<big>`большой`</big>` текст и `<small>`маленький`</small>` выделяются на фоне обычного

Вот что мы получим в результате:

большой текст и маленький выделяются на фоне обычного

Если взять стандартный текст за 100%, то тэг `<big>` увеличивает эту мерку до 110%, а тэг `<small>` уменьшает до 90%.

Верхние и нижние индексы

В последних версиях HTML появилось еще два новых тэга - тэги нижнего и верхнего индексов. Данные тэги удобны, когда вам нужно отобразить в своей странице такую специфическую информацию как химические формулы (H_2O) или математические формулы ($a^2=b+c^2$).

Чтобы текст выводился в виде нижнего индекса, его следует поместить между тэгами `<sub>` и `</sub>`, в виде верхнего - `<sup>` и `</sup>`. Например:

Химическая формула воды - $H_{2}O$

Используемая нами формула - $a^{2}=b+c^{2}$

Вот что мы получим в результате:

Химическая формула воды - H_2O

Используемая нами формула - $a^2=b+c^2$

Тэг редактирования стиля текста

Тэг имеет дополнительные возможности для задания размера шрифта основного текста. Вместо определенного номера вы можете задать относительный размер шрифта, например +3 или -1. Браузер прибавляет (или вычитывает) заданное число к номеру шрифта, используемого по умолчанию (размер 3). На пример:

```
<font size=+3>текст заданный параметром +3</font>  
<font size=-2>текст заданный параметром -2</font>
```

Вот что мы получим в результате:

```
текст заданный параметром +3  
текст заданный параметром -2
```

Поскольку шрифт обычно имеет размер +3, первая строка приведенного кода будет выведена на экран шрифтом 6 (3+3=6), а вторая будет иметь размер 1 (3-2=1).

Часто относительное задание размеров используется для вывода первой буквы фразы более крупным шрифтом, чем остальное предложение. Этот метод позволяет привлечь внимание к определенному фрагменту веб-страницы.

Ниже даны примеры вывода первого символа более крупным шрифтом, чем остальное предложение:

```
<font size=5>П</font>примеры из книги<br>  
<font size=6>Г</font>лава <font size=6>4</font>
```

Вот что мы получим в результате:

```
Примеры из книги  
Глава 4
```

Вложенные тэги

Часто тэг применяется с вложениями вдруг в друга для выделения определенных участков текста. Ниже приведен пример **не**правильного вложения тэга с параметром «size»:

```
<font size=+1>Крупный,  
<font size=+1>Крупнее,  
<font size=+1>Крупнейший</font></font></font>.
```

Вот что получится в результате:

Крупный, Крупнее, Крупнейший.

Логически было бы считать, что каждый раз прописывая увеличение размера текста приведет к суммированию размеров, то есть постепенному увеличению размера текста, но браузер понимает данный синтаксис совсем иначе. Тэги `<font>` не являются кумулятивными. Каждый тэг `<font size=+1>` задает увеличение размера до $3+1=4$, но это не говорит о том, что последующие тэги будут следовать алгоритму $3+1+1=5 \rightarrow 3+1+1+1=6$.

Чтобы последовательно увеличить размер шрифта, следует записать HTML-код следующим образом:

```
<font size=+1>Крупный</font>,  
<font size=+2>Крупнее</font>,  
<font size=+3>Крупнейший</font>.
```

Вот что получится в результате:

Крупный, Крупнее, Крупнейший.

Шрифты различного начертания

Обычно на компьютере установлено около 37 шрифтов. Internet Explorer позволяет задавать начертание (гарнитуру) шрифта, которым текст будет выведен на экран, с помощью параметра тэга `<font>` «face». Воспользуйтесь ключевым словом **face**="...", и браузер попытается отобразить текст тем альтернативным шрифтом, который вы указали. Например:

```
<font face="Arial">Шрифт Arial</font><Br>  
<font face="Courier New">Шрифт Courier New</font><Br>  
<font face="Times">Шрифт Times</font><Br>  
<font face="MS Sans Serif">Шрифт MS Sans Serif</font><Br>
```

Вот что получится в результате:

Шрифт Arial
Шрифт Courier New
Шрифт Times
Шрифт MS Sans Serif

Этот фрагмент кода (`face="..."`) сообщает браузеру, что отмеченный тэгом `<font>` текст должен быть отображен шрифтами указанными внутри `face="..."` [Arial, Courier New, Times, MS Sans Serif].

Тэг `<font>` работает следующим образом. Сначала браузер выясняет, какой шрифт вы задали, – в нашем случае Arial (возьмем за пример первую строку). Затем он проверяет, установлен ли шрифт Arial на персональном компьютере посетителя веб-страницы. Если да, то браузер отображает текст, используя этот шрифт. Если по какой-либо причине не удастся найти шрифт Arial, браузер просто выводит текст стандартным шрифтом, как если бы тэга `<font face="Arial">` вообще не было.

При желании можно задать список шрифтов, которые браузер будет искать перед тем, как прибегнет к шрифту, заданному по умолчанию.

```
<font face="Arial, Courier New, Times, MS Sans Serif">
```

В этом случае, если браузер не обнаружит шрифт Arial, он будет искать шрифт Courier New; если и Courier New не будет установлен, он пойдет дальше...

Используя этот тэг для выделения фрагмента, помните, что такое выделение увидят лишь только те посетители страницы, на чьих компьютерах установлен соответствующий шрифт. И еще, некоторые шрифты имеют разный размер, это видно из таблицы на абзац выше. Это говорит о том, что нерациональное применение разных шрифтов некрасиво отразится на дизайне вашего сайта.

При использовании данного тэга важно правильно называть шрифты. Чтобы увидеть список шрифтов, установленных на вашем компьютере, вам нужно выполнить следующие действия: Пуск -> Настройка -> Панель управления -> Шрифты.

Управление цветом шрифта

Можно менять цвет любого фрагмента отображаемой информации. Вы можете сделать так, чтобы слово или предложение автоматически отображалось одним из миллионов различных оттенков. Вы также можете выбрать ваш собственный цвет, смешивая различные оттенки красного, зеленого и синего.

Использование параметра `color=".."`. тэга `<font>` дает возможность задавать один из нескольких цветов для отображения текста, например:

```
<font color="red">Красный</font>
```

Вот что получится в результате:

Красный

Другие способы изменения цвета

Менять стандартный цвет страницы можно с помощью ключевого тэга `<body>`.

Отображение специальных символов

Иногда возникает необходимость отображать специальные символы на странице. Такие символы как &, <, >, ®, ©, ™, используются при написании некоторых тэгов. Другие символы не содержатся в обычном алфавите. Кроме того, существуют символы, используемые в иностранных языках, но не содержащиеся в английском - например, чтобы правильно отобразить на веб-странице слово français вам понадобится набрать **franç**, и **ais**. Смотрим:

français

Вот что получится в результате:

français

Таким же образом можно вставлять любые символы.

Символ	HTML-тэг	Символ	HTML-тэг
<	<	>	>
&	&	"	"
©	©	®	®
™	™	¢	¢
£	£	¤	¤
¥	¥	§	§
¨	¨	ª	ª
«	«	¬	¬
—	¯	°	°
±	±	²	²
³	³	'	´
μ	µ	¶	¶
·	·	ç	ç
,	¸	¹	¹
º	º	»	»
¼	¼	½	½
¾	¾	¿	¿
À	À	Á	Á
Â	Â	Ã	Ã
Ä	Ä	Å	Å
Æ	Æ	È	È

É	É	Ê	Ê
Ë	Ë	Ì	Ì
Í	Í	Î	Î
Ï	Ï	Ð	Ð
Ñ	Ñ	Ò	Ò
Ó	Ó	Ô	Ô
Õ	Õ	Ö	Ö
×	×	Ø	Ø
Ù	Ù	Ú	Ú
Û	Û	Ü	Ü
Ý	Ý	Þ	Þ
ß	ß		

В приведенном выше списке отображена только часть существующих специальных символов.

Задание к лабораторной работе №2.

1. Добавить текстовую информацию на свою страницу о своих увлечениях, интересах, о том, как проводите свободное время.

2. Отформатировать занесенный вами текст, используя тэги стилевого оформления текста, а так же при помощи тэга и специальных символов.

Урок HTML № 3

Списки и таблицы

Сегодня списки можно увидеть почти на каждой веб-странице. Существует несколько форматов списков, позволяющих выделить определенные фрагменты информации в стандартном тексте. Пункты списка отступают от полей, отделены от остальных абзацев текста и, кроме того, часто маркированы или нумерованы.

При правильном использовании списки делают большие объемы информации легко читаемыми и хорошо заметными. Списки могут быть вложены друг в друга, образуя сложную структуру. Вы можете, например, использовать список для перечисления ваших интересов или для организации ссылок.

Таблицы содержат информацию, расположенную по строкам и столбцам, и играют большую роль в организации веб-страницы. Они отображаются практически всеми веб-браузерами.

Таблицы очень удобны для представления большого количества связанной между собой информации в формате, который легко воспринимается и хорошо располагается на странице. Все, что можно организо-

вать в столбцы, стоит представить в виде таблицы. Компании часто используют таблицы для представления информации о товарах и ценах.

Каким образом списки помогают отформатировать текст, смотрим ниже:

Несмотря на то что я учусь, у меня остается довольно много свободного времени. У меня есть целый ряд интересов. Я работаю с программным обеспечением компании Microsoft. Кроме использования п/о компании Microsoft я использую программу Far Manager для обработки HTML-данных. Одно из моих любимых занятий – бродить по интернету, выискивая полезные ресурсы. Еще мне нравится слушать музыку группы Prodigy и песни исландской певицы Björk.

Этот абзац был слишком длинным. Кроме того, информация в нем повторялась и была плохо организована. Ниже приведен отрывок HTML-кода, показывающий, как можно его привести в порядок с помощью списков.

Мои интересы:

```
<ul>
<li> Програмное обеспечение от Microsoft и Far Manager
<li> Обзорение WWW
<li> Слушаю Prodigy и Björk
</ul>
```

Вот что получится в результате:

Мои интересы:

- Програмное обеспечение от Microsoft и Far Manager
- Обзорение WWW
- Слушаю Prodigy и Björk

В итоге проведенного форматирования информации, текст стал более наглядный и читабельный.

Типы списков

Существует три основных типа списков: маркированные, нумерованные и списки определений. Есть и другие типы, но эти три наиболее часто используются. Они сходны в том, что все они располагают каждый пункт на новой строчке, и вы можете выделить нужный текст. Главное различие перечисленных типов состоит в способе нумерации и структуре.

Создать список довольно легко. Сначала введите тэги открытия и закрытия списка:

```
<ul>  
</ul>
```

Затем перед текстом каждого пункта списка введите `<li>`:

```
<ul>  
<li> первая строка  
<li> вторая строка  
<li> третья строка  
</ul>
```

И теперь, вы можете озаглавить ваш список тэгами `<lh>` и `</lh>`:

```
<ul>  
<lh>Заголовок</lh>  
<li> первая строка  
<li> вторая строка  
<li> третья строка  
</ul>
```

Вот что получится в результате:

```
Заголовок  
• первая строка  
• вторая строка  
• третья строка
```

Нумерованный список

Нумерованный список очень напоминает маркированный. Единственным отличием является то, что в нумерованном списке перед каждым пунктом вместо графического маркера (точки) автоматически помещаются последовательные номера или буквы. Вы вводите новый пункт в начале списка, а последующие пункты нумеруются сами, что очень удобно.

Нумерованный список задается следующей командой:

```
<ul type="1">  
</ul>
```

Ниже приведен пример кода нумерованного списка:

```
<ul type="1">  
<li> первая строка  
<li> вторая строка
```

```
<li> третья строка
</ul>
```

Вот что получится в результате:

- первая строка
- вторая строка
- третья строка

Нумерованные списки удобны для описания информации, которую вы хотите определенным образом ранжировать (например, как в хит-параде), или стадий процесса, которые должны следовать в определенном порядке. Помимо параметра **type="1"** существует еще ряд других параметров. Посмотрите на следующую таблицу:

<pre><ul type="1"> первая строка вторая строка третья строка </pre>	<pre><ul type="A"> первая строка вторая строка третья строка </pre>	<pre><ul type="a"> первая строка вторая строка третья строка </pre>	<pre><ul type="I"> первая строка вторая строка третья строка </pre>	<pre><ul type="i"> первая строка вторая строка третья строка </pre>
• первая строка • вторая строка • третья строка	A. первая строка B. вторая строка C. третья строка	a. первая строка b. вторая строка c. третья строка	первая строка вторая строка третья строка	первая строка вторая строка третья строка

В этой таблице представлено 5 html-кодов и их результаты. В первом случае мы видим простой нумерованный список с арабскими цифрами. Во втором – список маркированный заглавными латинскими буквами. В третьем – маркировка строчными латинскими буквами. В четвертом – маркировка большими римскими цифрами. В пятом – маленькие римские цифры.

Списки определений

Хотя списки определений используются не так часто, как маркированные или нумерованные списки, они также могут быть полезны в некоторых случаях. В списке определений для каждого пункта представляется не одна, а две строки, что может быть полезно, когда вы приводите дополнительную информацию.

Списки определений выглядят несколько иначе, чем другие типы списков. Для описания каждого пункта списка в этом случае применяется два тэга, а не один тэг <i>. Тэг <dt> задает отдельный элемент, то есть определяемый термин, а тэг <dd> - остальную информацию, которая выводится на следующей строке с отступом. Вторая строка информации называется определением.

Идеальное использование списков такого типа это словарь. С помощью HTML вы можете легко задать каждый термин и его определение. В нижеприведенном примере первой строчкой списка будет название музыкальной команды, а второй - краткий комментарий.

Мои любимые музыкальные команды:

```
<ul>
<dt><b>Prodigy</b>
<dd><i>Группа родилась в 90-х годах. С самого начала состояла из 4-х
участников: Lime, Keith, Maxim и Leerooy. В 2000 году Leerooy решил
покинуть группу и начать сольную карьеру...</i>
<dt><b>Bjork</b>
<dd><i>Девушка начала заниматься музыкой в очень раннем возрасте.
Она работала совместно с SugarCubes, но быстро ушла из команды: сразу
после выхода неудавшегося их 2го совместного альбома...</i>
</ul>
```

Вот что мы получим в результате:

Мои любимые музыкальные команды:

Prodigy

Группа родилась в 90-х годах. С самого начала состояла из 4-х участников: Lime, Keith, Maxim и Leerooy. В 2000 году Leerooy решил покинуть группу и начать сольную карьеру...

Bjork

Девушка начала заниматься музыкой в очень раннем возрасте. Она работала совместно с SugarCubes, но быстро ушла из команды: сразу после выхода неудавшегося их 2го совместного альбома...

Списки в списках

Одно из свойств списков – это возможность вкладывать их друг в друга. Вложение списков в списки позволяет создать несколько уровней организации информации. Можно использовать несколько уровней списка на одной странице, что поможет упорядочить информацию

Вложение списка в список производится так же, как простое создание списка. Для этого не существует специальных тэгов. Чтобы не запутать браузер, убедитесь, что вы закрыли каждый внутренний список с помощью тэга `</ul>`. Вкладывать друг в друга можно даже типы списков (простой маркированный, нумерованный или список определений). Ниже приведен пример списка в списке:

```
Мои интересы:
<ul>
<li>Любимое п/о
  <ul>
    <li>Microsoft
    <li>Far Manager
  </ul>
<li>Любимые музыкальные исполнители
  <ul>
    <li>Prodigy
    <li>Björk
  </ul>
</ul>
```

Вот что получится в результате:

```
Мои интересы:
• Любимое п/о
  ○ Microsoft
  ○ Far Manager
• Любимые музыкальные исполнители
  ○ Prodigy
  ○ Bjork
```

Задание к 3 лабораторной работе:

1. Добавить на страницу список дел на неделю вперед, а также 3 самых любимых фильма и краткое их содержание.

2. Отформатировать информацию о делах на следующую неделю в виде двухуровневого списка (по дням недели).

3. Отформатировать информацию о любимых фильмах в виде списка определений.

Урок HTML № 4

Таблицы

У списков есть один недостаток – они одномерны. Это значит, что вы можете располагать информацию только на следующих друг за другом строках. Таблицы же позволяют располагать данные не только по строкам, но и по столбцам. У вас появляется гибкость двухмерной структуры для вывода информации на веб-страницу.

Важно использовать таблицы так, чтобы они не занимали попусту место на веб-странице. Благодаря наличию строк и столбцов таблицы удобно для сравнения и противопоставления информации. Каждые строку и столбец можно озаглавить, чтобы информация выглядела подобно электронной таблице.

Хорошая таблица делает веб-страницу информативной, аккуратной и организованной, а неудачная разбивает ее и запутывает то, что вы хотели сообщить. Ниже приведена таблица с понятной и четкой структурой.

Имя	Команда	Победы/поражения	Очки	Ауты
Пит Шурек	Цинциннати	16-2	2,73	194
Гидео Немо	Лос-Анджелес	19-8	2,94	399
Грег Энгриус	Атланта	24-6	2,21	275

Задание таблицы может показаться сложным, поскольку для этого применяется целый ряд тэгов. Тэги `<table>` и `</table>` обрамляют таблицу целиком, а ряд других тэгов определяет, каким образом будет выводиться информация. В следующей таблице приведем полное описание всех тэгов таблиц.

Тэги	Описание
<code><table></code> <code></table></code>	и Эти тэги охватывают таблицу. Тэг <code><table></code> сообщает браузеру, что далее следует описание таблицы. Если вы хотите, чтобы была видна сетка, разделяющая строки и столбцы, добавьте ключевое слово <code>BORDER</code> (получится <code><table border></code>), но подробнее об этом позже.
<code><caption></code> <code></caption></code>	и Текст, отмеченный этими тэгами, выводится в виде названия таблицы. Для задания названия таблицы можно также применять тэги <code><tc></code> и <code></tc></code> .

- `<th>` и `</th>` Тэги отображают текст заголовка или строки столбца немного более крупным полужирным шрифтом.
- `<tr>` и `</tr>` Тэги, определяющие каждую строку таблицы. Тэг `</tr>` обязателен, но делает html-код более красивым и понятным.
- `<td>` и `</td>` Эта пара тэгов определяет текст каждой ячейки таблицы.

При использовании этих тэгов следует быть очень аккуратным, иначе создание таблицы, состоящей из нескольких строк, может оказаться сложным делом. Чтобы создать даже такую простую таблицу, как была приведена в качестве примера чуть выше (про бейсбол), понадобится довольно много строк html-кода. Теперь покажем, в качестве примера правильной таблицы, код той самой таблицы:

```
<table border="1">
<tr>
  <th align="center">Имя</th>
  <th align="center">Команда</th>
  <th align="center">Победы/поражения</th>
  <th align="center">Очки</th>
  <th align="center">Ауты</th>
</tr>
<tr>
  <td>Пит Шупек</td>
  <td>Цинциннати</td>
  <td>16-2</td>
  <td>2,73</td>
  <td>194</td>
</tr>
<tr>
  <td>Гидео Немо</td>
  <td>Лос-Анджелес</td>
  <td>19-8</td>
  <td>2,94</td>
  <td>399</td>
</tr>
<tr>
  <td>Грег Энгриус</td>
  <td>Атланта</td>
  <td>24-6</td>
  <td>2,21</td>
  <td>275</td>
</tr>
</table>
```

Первый шаг – ввод инициализационного тэга таблицы:

```
<table>  
</table>
```

Чтобы ячейки таблицы были разделены тонкими линиями, добавим параметр **border** . Без этого слова ячейки будут автоматически выровнены, но сетки в таблице не будет. Теперь начнем создавать ячейки, одну за другой. Сначала зададим строку:

```
<table border="1">  
<tr>  
</tr>  
</table>
```

После этого зададим ячейки. Их у нас 5, соответственно 5 открывающих и закрывающих тэгов `<td>` и `</td>`

```
<table border="1">  
<tr>  
  <th>Имя</th>  
  <th>Команда</th>  
  <th>Победы/поражения</th>  
  <th>Очки</th>  
  <th>Ауты</th>  
</tr>  
</table>
```

Вот что получится в результате:

Имя	Команда	Победы/поражения	Очки	Ауты
-----	---------	------------------	------	------

Далее произведем центровку текста в таблице.

```
<table align="center" border="1">  
<tr>  
  <th>Имя</th>  
  <th>Команда</th>  
  <th>Победы/поражения</th>  
  <th>Очки</th>  
  <th>Ауты</th>  
</tr>  
</table>
```

Вот что получится в результате:

Имя	Команда	Победы/поражения	Очки	Ауты
-----	---------	------------------	------	------

Продолжим добавлять информацию:

```
<table align="center" border="1">
<tr>
  <th>Имя</th>
  <th>Команда</th>
  <th>Победы/поражения</th>
  <th>Очки</th>
  <th>Ауты</th>
</tr>
<tr>
  <td>Пит Шурек</td>
  <td>Цинциннати</td>
  <td>16-2</td>
  <td>2,73</td>
  <td>194</td>
</tr>
</table>
```

Вот что получится в результате:

Имя	Команда	Победы/поражения	Очки	Ауты
Пит Шурек	Цинциннати	16-2	2,73	194

Далее таблица заполняется аналогично выше приведенному примеру.

Расширенные возможности форматирования таблиц

Современные браузеры предлагают несколько способов настройки таблиц. При использовании каждого из этих методов подразумевается, что вы аккуратно сосчитали количество строк и столбцов. В каждом из разделов вы увидите измененную версию простой нашей "бейсбольной" таблицы.

Объединение нескольких строк в одну ячейку

Если вы будете часто применять таблицы, то непременно столкнетесь с ситуацией, когда какая-либо ячейка должна занимать несколько строк таблицы. Здесь вам пригодится параметр ROWSPAN, который добавляется в тэг <td> заданной ячейки. Чтобы ячейка занимала две строки вместо одной, замените ваш код, например <td>текст вашей простой

ячейки</td> на **<td rowspan=2> расширенный текст</td>**. Теперь при отображении таблицы ячейка с ключевым словом ROWSPAN займет две строки. Так можно изменить бейсбольную таблицу:

```
<table align="center" border="1">
<tr>
  <th>Имя</th>
  <th>Команда</th>
  <th>Победы/поражения</th>
  <th>Очки</th>
  <th>Ауты</th>
</tr>
<tr>
  <td>Пит Шурек</td>
  <td rowspan=2>Цинциннати</td>
  <td>16-2</td>
  <td>2,73</td>
  <td>194</td>
</tr>
<tr>
  <td>Хосе Рио</td>
  <td>28-2</td>
  <td>1,92</td>
  <td>199</td>
</tr>
</table>
```

Вот что получится в результате:

Имя	Команда	Победы/поражения	Очки	Ауты
Пит Шурек	Цинциннати	16-2	2,73	194
Хосе Рио		28-2	1,92	199

Объединение нескольких столбцов в одну ячейку

Подобно тому, как ROWSPAN распространяет ячейку на несколько строк, COLSPAN «раздвигает» ее на несколько столбцов. С помощью ключевого слова COLSPAN вы можете распространить ячейку на любое количество столбцов.

```

<table align="center" border="1">
<tr>
<th colspan=2>Личная информация</th>
<th colspan=3>Статистика</th>
</tr>
<tr>
<th>Имя</th>
<th>Команда</th>
<th>Победы/поражения</th>
<th>Очки</th>
<th>Ауты</th>
</tr>
<tr>
<td>Пит Шурек</td>
<td>Цинциннати</td>
<td>16-2</td>
<td>2,73</td>
<td>194</td>
</tr>
<tr>
<td>Гидео Немо</td>
<td>Лос-Анджелес</td>
<td>19-8</td>
<td>2,94</td>
<td>399</td>
</tr>
<tr>
<td>Грег Энгриус</td>
<td>Атланта</td>
<td>24-6</td>
<td>2,21</td>
<td>275</td>
</tr>
</table>

```

Личная информация		Статистика		
Имя	Команда	Победы/поражения	Очки	Ауты
Пит Шурек	Цинциннати	16-2	2,73	194
Гидео Немо	Лос-Анджелес	19-8	2,94	399
Грег Энгриус	Атланта	24-6	2,21	275

Включение списка в таблицу

Таблицы можно комбинировать с другими элементами HTML. Так, они могут содержать списки любого типов. Строго следите за тем, чтобы всегда использовались закрывающие тэги, иначе коды списков трудно отслеживать. Ниже представлен пример включения в таблицу маркированного списка:

```
<table align="center" border="1">
<tr>
  <th>Имя</th>
  <th>Команда</th>
  <th>Победы/поражения</th>
  <th>Очки</th>
  <th>Ауты</th>
</tr>
<tr>
  <td>Пит Шупек</td>
  <td>Цинциннати</td>
  <td>16-2</td>
  <td>2,73</td>
  <td>194</td>
  <td>
    <ul>
      <li>команда "All-Star Team"</li>
      <li>Награда MVP</li>
      <li>Серебряная летучая мышь</li>
    </ul>
  </td>
</tr>
<tr>
  <td>Гидео Немо</td>
  <td>Лос-Анджелес</td>
  <td>19-8</td>
  <td>2,94</td>
  <td>399</td>
</tr>
<tr>
  <td>Грег Энгриус</td>
  <td>Атланта</td>
  <td>24-6</td>
  <td>2,21</td>
  <td>275</td>
</tr>
</table>
```

Имя	Команда	Победы/поражения	Очки	Ауты	
Пит Шурек	Цинциннати	16-2	2,73	194	• команда "All-Star Team" • Награда MVP • Серебряная летучая мышь
Гидео Немо	Лос-Анджелес	19-8	2,94	399	
Грег Энгриус	Атланта	24-6	2,21	275	

Выравнивание текста в таблице

Для каждой ячейки вы можете задать способ выравнивания текста, как по вертикали, так и по горизонтали. Способы выравнивания задаются специальными ключевыми словами **valign** и **align**. Вы можете задать также способ выравнивания для всей таблицы полностью или для заданной строки. Но, как правило, это выполняется только для одной ячейки.

Каждое из ключевых слов **align** и **valign** имеет четыре возможных значения.

Значение ключевого слова

Описание

ALIGN=LEFT	Выравнивает текст в ячейке (таблице, строке) по левому краю (идет по умолчанию)
ALIGN=CENTER	Выравнивает текст в ячейке (таблице, строке) по центру
ALIGN=RIGHT	Выравнивает текст в ячейке (таблице, строке) по правому краю
ALIGN=JUSTIFY	Растягивает текст в ячейке (таблице, строке) на всю строку
VALIGN=TOP	Растягивает текст от самого верха ячейки (полезно, когда информация в строке состоит из нескольких строк)
VALIGN=MIDDLE	Выравнивает текст в ячейке в ячейке вертикально по центру (идет по умолчанию)
VALIGN=BOTTOM	Выравнивает текст в ячейке по нижнему краю

Продолжая работать с бейсбольной таблицей (та, в которой есть список), введем ключевое слово VALIGN=TOP для всей строки, содержащей список, чтобы текст не "плавал" в центре ячейки. Изменим эту строку таким образом:

```
<table align="center" border="1">
<tr>
  <th>Имя</th>
  <th>Команда</th>
  <th>Победы/поражения</th>
  <th>Очки</th>
  <th>Ауты</th>
</tr>
<tr valign=top>
  <td>Пит Шурек</td>
  <td>Цинциннати</td>
  <td>16-2</td>
  <td>2,73</td>
  <td>194</td>
  <td>
    <li>команда "All-Star Team"
    <li>Награда MVP
    <li>Серебряная летучая мышь
  </td>
</tr>
<tr>
  <td>Гидео Немо</td>
  <td>Лос-Анджелес</td>
  <td>19-8</td>
  <td>2,94</td>
  <td>399</td>
</tr>
<tr>
  <td>Грег Энгриус</td>
  <td>Атланта</td>
  <td>24-6</td>
  <td>2,21</td>
  <td>275</td>
</tr>
</table>
```

Вот что получится в результате:

Имя	Команда	Победы/ поражения	Очки	Ауты	
Пит Шурек	Цинциннати	16-2	2,73	194	- команда "All-Star Team" - Награда MVP - Серебряная летучая мышь
Гидео Немо	Лос-Анджелес	19-8	2,94	399	
Грег Энгриус	Атланта	24-6	2,21	275	

Цвета в таблице

Существует ряд тэгов, позволяющих задать цвета фона и сетки таблицы. В первом случае ключевое слово **bgcolor** вставляется в тэг <table> следующим образом:

```
<table bgcolor="yellow">
</table>
```

Кроме **bgcolor**, существуют другие способы управления цветом:

Значение ключевого слова

BORDERCOLOR="..."

Описание

Изменяет цвет сетки таблицы

BORDERCOLORDARK="..." /
BORDERCOLORLIGHT="..."

Используется для изменения цвета сетки с дополнительным эффектом трехмерности. Если установлены эти атрибуты, ваша таблица выглядит трехмерной.

Имя	Команда	Победы/поражения	Очки	Ауты	
Пит Шурек	Цинциннати	16-2	2,73	194	- команда "All-Star Team" - награда MVP

					• Серебряная летучая мышь
Гидео Немо	Лос-Анджелес	19-8	2,94	399	
Грег Энгриус	Атланта	24-6	2,21	275	

Такого вида таблицы мы добились лишь с помощью этих самых ключевых тэгов:

```

<table align="center" bordercolordark=navy bordercolorlight=blue
border="1">
<tr bgcolor="yellow">
    <th>Имя</th>
    <th>Команда</th>
    <th>Победы/поражения</th>
    <th>Очки</th>
    <th>Ауты</th>
</tr>
<tr valign=top bgcolor="#abcdef">
    <td>Пит Шупек</td>
    <td>Цинциннати</td>
    <td>16-2</td>
    <td>2,73</td>
    <td>194</td>
    <td bgcolor="#bcdefa">
        <li>команда "All-Star Team"
        <li>награда MVP
        <li>Серебряная летучая мышь
    </td>
</tr>
<tr bgcolor="#abcdef">
    <td>Гидео Немо</td>
    <td>Лос-Анджелес</td>
    <td>19-8</td>
    <td>2,94</td>
    <td>399</td>
</tr>
<tr bgcolor="#abcdef">
    <td>Грег Энгриус</td>

```

```

<td>Атланта</td>
<td>24-6</td>
<td>2,21</td>
<td>275</td>
</tr>
</table>

```

Альтернативы таблицам

Существуют альтернативные способы создания таблиц, применяются только в случае использования старых браузеров.

Вместо таблицы используется список

Несмотря на то, что списки "одномерны", порой списков при надлежащем использовании можно заменить практически любую таблицу. Например, вы можете заменить бейсбольную таблицу несколькими списками. Конечно, эти списки не так легко читать, как таблицу, и они заставляют пользователя перемещаться по экрану. В общем случае списки менее компактны, чем таблицы. Кроме того, информацию в списках труднее сравнивать. Ниже приведен список, заменяющий таблицу:

- Пит Шурек
 - Цинциннати
 - 16-2 (W/L)
 - 2.73 очков
 - 194 аута
 - команда "All-Star Team"
 - Награда MVP
 - Серебряная летучая мышь
- Гидео Немо
 - Лос-Анджелес
 - 19-8 (подеды/поражения)
 - 2.94 очков
 - 399 аута
- Грег Энгриус
 - Атланта
 - 24-6 (победы/поражения)
 - 2.21 очков
 - 275 аутов

Предварительное форматирование

При использовании тэгов `<pre>` и `</pre>` информация отображается браузером точно в таком виде, в каком вы ее напечатаете, без всякой интерпретации. Вы можете воспользоваться этими тэгами для эмуляции таблицы. Результат не так эффектен, как таблица, и здесь невозможно

ввести сетку таблицы, но большинство людей не заметит особой разницы. Предварительно отформатировав текст с использованием возвратов каретки, пробелов и табуляций так представит все ту же бейсбольную информацию:

```
<PRE><B>
Имя      Команда  Победы/поражения Очки Ауты</B>
Джон Хеппи  Цинциннати  16-2      2.73 194
Гидео Немо  Лос-Анджелес 19-8      2.94 399
Грег Энгриус Атланта   24-6      2.21 275
</PRE>
```

Вот что получится в результате:

Имя	Команда	Победы/поражения	Очки	Ауты
Джон Хеппи	Цинциннати	16-2	2.73	194
Гидео Немо	Лос-Анджелес	19-8	2.94	399
Грег Энгриус	Атланта	24-6	2.21	275

Задание к лабораторной работе № 4:

1. Разместить на своей странице свое расписание занятий на неделю в виде таблицы.
2. Отформатировать таблицу, чтобы она содержала:
 - 2.1. Название.
 - 2.2. Объединенные ячейки в строках и столбцах.
 - 2.3. Заголовков, с выравниванием по центру.
 - 2.4. Время занятий с выравниванием по левому краю и центру.
 - 2.5. Дни недели с выравниванием по низу ячейки и центру.
 - 2.6. Название предметов с выравниванием по центру.
 - 2.7. Ячейки в таблице должны быть объемными.
 - 2.8. Шапка таблицы должна содержать более темный фон, нежели остальная таблица.

Урок HTML № 5

Применения графики на веб-страницах

В наше время графическими изображениями украшены абсолютно все интернет-ресурсы. От элементарных логотипов фирм и названий интернет-ресурсов до рекламных изображений, предлагающих посетить тот или иной сайт, а также все возможные социальные сети, которые пестрят различными изображениями на любой цвет и вкус.

Вопрос о необходимости применения графики на веб-сайтах уже давно потерял актуальность, графику применять однозначно нужно. В наше время нужно задаваться другим вопросом, в каких масштабах ее стоит использовать. Если сайт слишком перегрузить информацией он потеряет наглядность и будет долго загружаться. Именно поэтому необходимо использовать графику на веб-страницах разумно.

Источники изображений

Изображения можно взять как с другого сайта, так создать самому, либо при помощи специальных программных средств по созданию графических объектов, либо при помощи сканера (просто отсканировав картинку, фото), либо при помощи цифрового фотоаппарата или телефона с фотокамерой.

Используемые форматы изображений

Изображения можно сохранять в нескольких форматах. У каждого из них есть свои преимущества и недостатки. Чаще всего используются - GIF и JPEG. Тип файлов GIF (Graphical Interchange Format - Графический Формат Обмена) был впервые применен много лет назад CompuServe (онлайновая информационная служба) для предоставления информации в стандартном графическом формате. Он стал первым типом файлов, поддерживаемых WWW.

Формат JPeG (Joint Photographic Experts Group - Объединенная Группа Экспертов Фотографии). Для изображений большого объема или более ярких и насыщенных используется формат JPEG, для небольших изображений реклам или просто пометок используется GIF.

Отображение изображения на странице

Имеется страница, на которой имеется пока только текст. Например, эта:

== Моя коллекция бабочек ==

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Стоит нам добавить одну строчку и страничка превращается в следующее:

== Моя коллекция бабочек ==



Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Посмотрите, что мы добавили:

```
<center>  
<h3>== Моя коллекция бабочек ==</h3>  
<br><br>  
</center>
```

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Изображения помещаются на странице крайне просто, использует-ся тэг `<img>`. IMG – сокращение от «image» («изображение»). Надо ввести тэг `<img>` и ключевое слово `SRC="..."` для указания, какую картинку загрузить. Изображения можно помещать в любом месте веб-страницы, в заголовки, таблицы, списки или в текст, только проверьте, чтобы тэг `<img>` оказался между тэгами `<body>` и `</body>`. В данной странице о бабочках мы собираемся ввести изображение бабочки, вверху страницы, как раз под заголовком `<h1></h1>`.

Мы вставляли здесь картинку бабочки в формате GIF, под названием **dc72.gif**, которая находится в папке с нашей интернет-страницей.

Тэг `<img>` сообщает браузеру, что необходимо вывести на экран изображение с именем исходного файла **dc72.gif**. Вводя этот тэг, вы подразумеваете, что файл **dc72.gif** находится в папке с нашей интернет-страницей.

Основываясь на информации, приведенной, легко определить, когда вы указали неверное расположение графического файла: браузер вместо изображения, которое вы ожидали увидеть, выводит вместо картинки пиктограмму ошибки такого вида:



Эта пиктограмма показывается браузером в случаях, если:

- Имя файла, содержащего изображение, задано неверно и не указывает на существующий файл.

- Неверно указан путь к файлу.
- Некоторые сервера чувствительны к регистру названий, то есть имя файла написана заглавными или прописными буквами.

Оптимальные размеры изображений

В зависимости от того какую информацию вы собираетесь размещать на своем веб-сайте вам нужно будет выбирать и размер ваших изображений. Например, если вы создаете сайт-визитку, то одно из самых важных условий это скорость загрузки и максимальная информативность этих веб-страниц, соответственно вам нужно будет уменьшать размер ваших до минимума, но в то же время смотреть чтобы не слишком сильно портилось их качество. В то же время если вы создаете сайт, содержащий фотографии и выставку, то упор должен делаться на максимум качества изображения, хотя это и будет приводить к снижению скорости загрузки ваших страниц.

Альтернативный текст

Некоторые пользователи всемирной сети специально отключают автоматическое отображение картинок для более быстрой загрузки веб-страниц. И чтобы ваши картинки не полностью потеряли свою информативность предусмотрена возможность отображения альтернативного текста, при не отображении изображений, а также для отображения информативного текста при наведении указателя курсора на изображение, если оно отображается.

В описанных выше случаях полезно использовать ключевое слово ALT тэга . ALT выводит заданный текст в той части экрана, где должно быть (или будет) изображение.

Задание альтернативного текста, описывающего изображения домашней страницы, считается правилом вежливости. Альтернативный текст является частью тэга . Просто добавьте к тэгу ключевое слово ALT="..." и наберите текст в кавычках после знака "равно". Данный тэг с альтернативным текстом, к примеру с **dc72.gif**, выглядит так:

== Моя коллекция бабочек ==



Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Посмотрите, что мы добавили:

```
<center>  
<h3>-= Моя коллекция бабочек =-</h3>  
<br><br>  
</center>
```

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Выравнивание изображений

Для размещения изображения на странице используется параметр align тэга

Опция	Действие
LEFT	Выравнивает изображение по левому краю строки, строки текста выводятся справа от изображения
RIGHT	Действует подобно LEFT, но выравнивает изображение по правому краю строки
TOP	Выравнивает изображение по самому высокому элементу в строке.
TEXTTOP	Выравнивает изображение по самому высокому элементу в строке (обычно совпадает с top)
MIDDLE	Выравнивает нижнюю границу строки посередине изображения
ABSMIDDLE	Выравнивает середину строки посередине изображения (очень похожа на MIDDLE, но применяется для мелких изображений)
BASELINE	см. сл.
BOTTOM	Выравнивает нижнюю границу строки по нижней границе изображения

```
<center>  
  
<h3>-= Моя коллекция бабочек =-</h3>  
</center>
```

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Посмотрите, что получится:



== Моя коллекция бабочек ==

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Задание размера изображения

Вы можете управлять не только способом выравнивания изображения, но и его высотой и шириной. Как правило, изображения выводятся в натуральную величину, но с помощью параметров тэга `<img>` HEIGHT и WIDTH можно сжать или растянуть изображение, не изменяя самой картинку. Кроме того, ключевые слова HEIGHT и WIDTH позволяют отвести вполне определенное место под изображение и пустить текст вокруг него обложкой. В этом случае, пока загружаются изображения, остальная страница будет уже видна. Значения HEIGHT и WIDTH задаются в пикселях. Тем самым задается доля размеров картинки по отношению к размерам экрана.

Ключевые слова HEIGHT и WIDTH используются в тэге `<img>` наряду с ключевыми словами выравнивания и альтернативного текста. Когда мы ввели ключевое слово HEIGHT и WIDTH в тэг картинки, получился следующий код:

```
<center>  
  
<h3>== Моя коллекция бабочек ==</h3>  
</center>
```

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Посмотрите, что получится:



== Моя коллекция бабочек ==

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Теперь изображение бабочки на экране размерами в 100x100. HEIGHT – высота, WIDTH – ширина.

Пиктограммы

Кроме полноцветных изображений (подобной бабочке) и рисунков, на странице могут присутствовать пиктограммы всех видов. С технической точки зрения пиктограммы попадают в категорию "изображения" (они обычно хранятся в файлах формата GIF или JPeG), но обычно очень малы 0.5-5kb, и применяются на домашних страницах из соображений дизайна, эстетики, а также для «перелистывания» страниц.

Существуют самые различные пиктограммы. На страницах WWW вы найдете их великое множество, начиная от миниатюрных пиктограмм реконструкции до цветных линий и кнопок. Иногда при просмотре домашних страниц вы даже не отдаёте себе отчёта в том, что вы смотрите на пиктограммы, насколько хорошо они интегрированы в страницу.

Линии и полосы

Раньше мы говорили о применении тэга <hr> для того, чтобы разделить веб-страницу на части линиями. Очень часто вместо тэга <hr> используют линейную графику (простую графику в виде полос или линий).

Элементы линейной графики бывают различных форм, цветов и рисунков и существенно отличаются от результата работы тэга <hr>. Например, на нашей странице необходимо нужно поставить красивый разделитель под названием **34499.gif**:

```
<center>

<h3>== Моя коллекция бабочек ==</h3>
<br><br>
</center>
```

Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Посмотрите, что получится:



== Моя коллекция бабочек ==



Я коллекционирую бабочки. У меня очень большая коллекция, примерно 1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабочек у меня различных видов и окраски.

Фон страниц

Одна из возможностей HTML – это задание фона документа. Вместо того чтобы создавать страницы со стандартным серым фоном, вы можете задать любой другой цвет. Если даже вам не нравится однотонный фон, вы можете задать фоновый узор.

Параметр BGCOLOR тэга <body> отвечает за изменение всего фона вашей страницы, существует 16'777'216 цветов фона.

Вы можете задавать не только стандартные цвета, но и фоновые рисунки. Текст и изображение, естественно, располагаются поверх фона.

Допустим, у нас есть фоновая картинка **967_24.jpg** расположенная на **<http://i.smartphone.ua/img/wp>**. Посмотрите, как с помощью ключевого слова BACKGROUND="..." мы поставим эту картинку как фон в нашей странице:

```
<body background="http://i.smartphone.ua/img/wp/967_24.jpg ">
<center>

<h3>-= Моя коллекция бабочек =-</h3>
<br><br>
</center>
Я коллекционирую бабочки. У меня очень большая коллекция, примерно
1300 различных бабочек. Я стараюсь не повторяться и около 1000 бабо-
чек у меня различных видов и окраски.<br><br>
</body>
```

Задание к лабораторной работе № 5

1. Добавить на вашу страницу изображения:
 - 1.1. Изображения ваших увлечений, таким образом, чтобы изображения увлечений находились справа от их названий.
 - 1.2. Напротив информации о ваших любимых фильмах разместить изображения-плакаты, иллюстрирующие фильмы.
 - 1.3. В таблице с расписанием добавить еще одну строку в начало таблицы и разместить там логотип МГТУ.
 - 1.4. Создать у вашей страницы фон-изображение.
 - 1.5. Добавить другие изображения на ваше усмотрение для придания вашей странице соответствующего дизайна и уникальности.

Урок HTML № 6

Ссылки на веб-страницах

Гиперссылки на веб-страницы – одно из основных свойств WWW. Любой документ может содержать ссылку на другой веб-документ, заданную с помощью специального тэга в HTML.

Гипертекстовые ссылки – это фрагменты текста, часто подчеркнутые и выведенные синим текстом. Отображенный таким образом текст называется текстом ссылки. Если на нем кликнуть мышью, он автоматически осуществляет переход на другую веб-страницу.

В каждом URL-адресе есть три части: протокол, узел Интернета и имя файла (включая путь нему, если необходимо). **Протокол** – это правила, в соответствии с которыми два компьютера общаются друг с другом; в этой главе говорится в основном о стандартном http (hyper text transfer protocol – протокол трансляции гипертекста, т.е. HTML) - стандартном протоколе WWW. Вы должны указать также узел Интернета и при необходимости имя файла.

Рассмотрим пример такого адреса: **http://dlab.by.ru/subscr/html.arc/ index.html**. Браузер должен с помощью определенного способа связи (http) связаться по Интернету с указанным адресом **dlab.by.ru** (сайт Design Laboratory), затем найти веб-страницу в поддиректории **subscr/html.arc/** и запустить **index.html**.

Гипертекстовые ссылки часто используются, поскольку могут легко связать два документа, расположенные на противоположных концах мира. Каждый документ и файл в Интернете имеют уникальный адрес, называемый URL'ом (uniform resource locator - универсальный указатель ресурса). Чтобы связать два документа, разработчики веб-страниц помещают на одном из них URL другой.

Независимо от того, где находятся веб-страница, на которые вы хотите сослаться – в России или в Новой Зеландии, – браузер будет работать с ними одинаково. Браузер использует каждый URL для того, чтобы найти документ в Интернете и автоматически доставить его вам. Тому, кто просматривает веб-узлы по всему миру, нет нужды думать ни о URL'е, ни о том, как найти нужный документ. Обо всем этом позаботится ваш браузер.

С помощью WWW вы можете непосредственно на веб-странице ссылаться на HTML-документы (веб-страницы), файлы (по протоколу FTP [file transfer protocol – протокол передачи файлов]), группы новостей Интернета (например, UseNet) и даже на такие популярные источники информации как Yandex или Yahoo!.

Ссылки можно создавать на веб-странице только по отдельности. Вам необходимо определить, что будет служить текстом ссылки и, еще важнее, какую ссылку будет представлять каждый такой текст. Вы може-

те поместить на веб-странице сколько угодно ссылок и скомпоновать их любым способом.

Ссылаться на другие страницы достаточно просто, но при этом необходимо понимать, каким образом помещать ссылки на странице, а также знать соответствующий синтаксис HTML.

Как любой элемент HTML, ссылка задается парой тэгов. Открывающий тэг (называемый якорем) - это тэг <a>. Затем вы задаете файл, на который ссылаетесь, а также текст ссылки. Все это завершается закрывающим тэгом .

Вот так, например, выглядит HTML-код ссылки, на сайт Design Laboratory:

```
<a href="http://ertu.wsdu.pl/~Sam/dlab">Design Laboratory</a>
```

<a> - якорь; href="http://ertu.wsdu.pl/~Sam/dlab" - ссылка; Design Laboratory – текст ссылки; - закрытие якоря.

Создание ссылки на местные веб-страницы

Вполне вероятно, что в одном месте у вас окажется несколько HTML-документов. Допустим, у вас слишком много информации для одного документа, и вы решили разбить его на несколько HTML-файлов. В таком случае естественно поместить на главной странице ссылки на каждый из этих документов, которые позволят легко и быстро на них перейти.

Рассмотрим процесс создания небольшого оглавления. В первую очередь необходимо уточнить названия ваших интернет-страниц и каким образом будут называться ссылки.

Текст ссылки	Имя файла
Обо мне	about.html
полезные сайты	links.html

Теперь введите тэг <a href>. Тэг <a> называют якорем. Он применяется для ссылок на другие веб-страницы. Ключевое слово HREF якорного тэга сообщает браузеру, на что вы ссылаетесь. Не забудьте про закрывающий тэг . Чтобы создать ссылку на about.html нужно ввести следующий код:

```
<a href="about.html">Обо мне</a>
```

Фраза «Обо мне» появится на экране в виде текста ссылки. Когда посетители, кликнув на этом тексте, они откроют файл about.html.

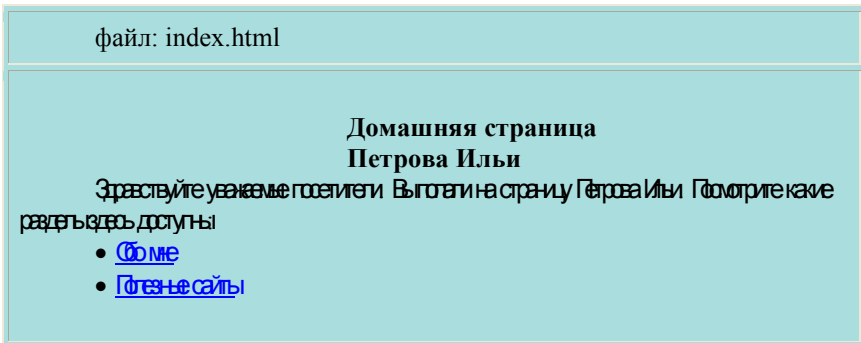
Проверьте, не забыли ли вы, если это необходимо, указать путь к файлам, на которые ссылаетесь. Например, если бы файл about.html был сохранен в папке INFO, мы бы задали с ключевым словом HREF путь info/about.html:

Обо мне

Сейчас на вашей страничке появится только текст «Обо мне». Выполнив ту же процедуру, мы добавляем еще одну ссылку, посвященную полезным сайтам:

Полезные сайты

Действие ссылки можно проверить, кликнув на ней и посмотрев, попали ли вы на нужную страницу. До того как вы кликните на ссылке, браузер выведет полный URL того документа, который вы хотите посетить, в строке состояния внизу экрана. Давайте посмотрим на нашу страничку ссылками:



Ссылки на любое место в WWW

Для ссылки на документы, расположенные не на вашем веб-узле необходимо знать полный URL документа. Тем не менее, ссылка выглядит так же. Вы снова обозначаете, имя файла и текст ссылки тэгами **** и ****. Когда вы ссылаетесь на другой HTML в Web, URL всегда начинается с **HTTP://**.

Таким образом браузер узнает, что искать следует документ в WWW, а не файл на вашем веб-узле. Остальная часть URL'a состоит из пути в Интернет к нужному узлу и имени файла документа, который вы ищете. Это очень похоже на написание адреса на конверте: чтобы адресата можно было найти, вы должны правильно указать все части его адреса.

В качестве иллюстрации ссылки на любое место в Интернете мы поместим на страницу ссылку на список веб-страниц цирков на поисковике [Yahoo!](#). Первый шаг – определение полного URL'a, необходимо для доступа к документу. Следующий – задание текста, осуществляющего ссылку в окне браузера. Мне понадобится URL списка цирков в поисковике [Yahoo!](#). Через каталог мы пойдем по следующему пути: Arts (искусство) → Performing Arts (представления) → Circuses (цирки). Теперь в адресной строке (та, что вверху браузера) мы видим URL страницы цирков. Он таков:

http://www.yahoo.com/arts/performing_arts/circuses/

Теперь на своей страничке я введу в тэги `<a href>` и `</a>` адрес списка цирков в [Yahoo!](http://www.yahoo.com/arts/performing_arts/circuses/):

`<a href="http://www.yahoo.com/arts/performing_arts/circuses/"`
Список цирков в поисковике Yahoo!`</a>`

После того как ссылки помещены на странице, можно легко перемещаться с одной страницы на другую. Обратите внимание, что, когда указатель мыши попадает на гипертекстовую ссылку, браузер выводит в строке состояния внизу окна браузера полный URL того места, на которые вы хотите перейти.

Если вы не хотите, чтобы при переходе на другую страницу закрывалась ваша добавте в тэге `<a href="...">` параметр `target="_blank"`, ссылка откроется в новом окне браузера. Т.е. сначала будет открыто параллельное окно браузера, а затем это окно откроет URL на который вы ссылались. Посмотрите на пример, который откроет URL на список цирков в Yahoo! в новом окне:

`< target= "_blank"`
`href="http://www.yahoo.com/arts/performing_arts/circuses/"` **Список**
цирков в поисковике Yahoo!`</a>`

Цвет ссылки

Вы можете изменить цвет текста ссылки. Мы уже говорили, как заменять цвета простого текста с помощью тэга `<body>`. Можно заменить цвет ссылки еще и тэгом `<font>`. Посмотрите:

`<a href="http://www.rambler.ru"><font color="green">` Рамблер
`</font></a>`

Поглядите, как будет выглядеть такая ссылка на практике:

[Рамблер](http://www.rambler.ru)

Использование изображений для ссылок

В качестве средства ссылок на другие веб-страницы, можно использовать не только текст, но и изображения. Так же, как вы ссылались на другой документ с помощью текста, вы можете сослаться на него с помощью изображения. Нужно только вместо текста ссылки задать изображение. По умолчанию браузеры обнаруживают изображение с синей рамкой. Кликнув мышью в любом месте этого изображения, посетители переходят на соответствующий документ. В качестве ссылок на веб-документы вы можете использовать любые виды изображений, рисунков и пиктограмм.

Очень популярны также изображения-карты. В качестве изображения-карты может применяться веб-графика любого типа. Такое изоб-

ражение открывает различные файлы в зависимости от того, на каком фрагменте был произведен щелчок мышью. На следующих уроках вы узнаете об этом поподробнее.

Давайте представим, что нужно вставить графический рисунок Design Laboratory на страничку. Посмотрите на это:



Как вы видите синего бордюра обрамляющего картинку нет. В тэг `<img>` мы добавили ключевое слово **`border="0"`**, дающее команду "отключить обрамление". Посмотрите теперь на код изображения:

```
<a href="http://ertu.wsdu.pl/~Sam/dlab"></a>
```

Обратите внимание, якорь остался тем же, он ссылается на сайт Design Laboratory, но вместо текста мы ввели тэг `<img>`.

Якоря

HTML-ссылки так же можно использовать в качестве внутренних ссылок и указателей.

Если у вас большой HTML-документ, в различных его местах нужно поставить якоря. Так, если бы толстая книга существовала в виде одного большого HTML-файла, логично было бы установить якорь в начале каждой главы. В начале HTML-файла нужно сделать оглавление, ссылающееся на каждый якорь в вашем файле (или на каждую главу). Если посетитель кликнет на якорь с меткой "глава 26", браузер автоматически перенесет его на якорь "глава 26" в том же файле; ему не придется просматривать бесчисленное количество страниц с информацией. Не забывайте, что якоря предназначены только для работы в одном большом HTML-файле. Чтобы организовать несколько файлов, применяйте гипертекстовые ссылки.

Якоря действуют так же, как и ссылки на другие документы в Интернете, только ссылки происходят на внутренние метки документа. Как и с обычными ссылками, вы можете использовать сколько угодно якорей и тэгов.

Разработчики веба часто называют применяемые таким образом якорные тэги флажками.

Создание якоря

Работа с якорными тэгами документа содержит два важных этапа. Во-первых, определите, на какие разделы HTML-документа вы хотите сослаться. Во-вторых, введите тэги `<a href="..">`, указывающие на эти разделы.

Вы можете вводить именные якоря в любом месте ваших веб-страниц. Каждый тэг позволяет непосредственно перейти в нужное место с помощью ссылки. В данном примере поместим четыре якорных тэга и соответствующие им ссылки.

```
<a href="#raz1">Переход к разделу 1</a><br>
<a href="#raz2">Переход к разделу 2</a><br>
<a href="#raz3">Переход к разделу 3</a><br>
<a href="#raz4">Переход к разделу 4</a><br>
<br>
<a name="raz1">Раздел 1</a><br>
Ваш текст<br>
...<br>
...<br>
...<br>
Ваш текст<br>
<a name="raz2">Раздел 2</a><br>
Ваш текст<br>
...<br>
...<br>
...<br>
Ваш текст<br>
<a name="raz3">Раздел 3</a><br>
Ваш текст<br>
...<br>
...<br>
...<br>
Ваш текст<br>
<a name="raz4">Раздел 4</a><br>
Ваш текст<br>
...<br>
...<br>
...<br>
Ваш текст<br>
```

"Переход к разделу 1" - этой строчкой мы обозначили якорным тэгом имя **Переход к разделу 1**. Чтобы перейти на этот тэг, посетитель должен кликнуть на флажке с именем **raz1**.

[Переход к разделу 1](#)

[Переход к разделу 2](#)

[Переход к разделу 3](#)

[Переход к разделу 4](#)

Раздел 1

Ваш текст

...

...

...

Ваш текст

Раздел 2

Ваш текст

...

...

...

Ваш текст

Раздел 3

Ваш текст

...

...

...

Ваш текст

Раздел 4

Ваш текст

...

...

...

Ваш текст

Ссылка на якорь

После того как вы установили все необходимые якоря, настала пора составить оглавление, содержащее ссылки на каждый из якорей.

Ссылка на якорь очень напоминает стандартную ссылку. Для этого используется тэг . Отличие в том, что вам надо сообщить браузеру, что это ссылка на якорь внутри документа. Это делается с помощью значка #. Для задания ссылки на раздел нашей «золотой» страничке, использовался тэг:

**Переход к разделу 4
**

Символ # указывает браузеру, что следует искать флажок, а не страничку. Для посетителей текст такой ссылки будет выглядеть так же, как и текст любой другой ссылки. Однако при клике на таком тексте он перейдет не на отдельный HTML-файл, а на якорь, расположенный в той же странице что и ссылка на него.

Компоновка ссылок в виде списков

Применяя на веб-странице ссылки, важно располагать их организованно, в определенном порядке, чтобы они были понятны, и ими было легко пользоваться. Каждый раз, добавив очередной фрагмент HTML, убедитесь, что страница осталась организованной и читаемой.

Ссылки часто komponуют в виде списка. Посмотрите на пример:

- [Design Laboratory](#) - сайт для веб-народа. Большая коллекция графики, софтины и прочих полезностей для вебмастеров.
- [WebMaster](#) - сайт для веб-народа. Коллекция клипартов, графики, скриптов и прочих инструментов.

Посмотрите на код:

```
<li><a href="http://ertu.wsdu.pl/~Sam/dlab">Design Laboratory</a>  
- сайт для веб-народа. Большая коллекция графики, софтины и прочих  
полезностей для вебмастеров.<br>  
<li><a href="http://gif.lgg.ru">WebMaster</a> - сайт для веб-народа.  
Коллекция клипартов, графики, скриптов и прочих инструментов.<br>
```

Задание к лабораторной работе № 6

1. Создать две html-страницы, в дополнение к уже существующей.
 - 1.1. На одной из них указать ссылки на полезные интернет ресурсы (поисковые системы, сборники статей и рефератов и любые другие интернет-ресурсы) в виде списка, с краткой характеристикой каждого источника.
 - 1.2. На второй странице расположить доклад о пользе компьютерной грамотности.
2. Разбить ваш доклад на несколько частей и сделать ссылочный доступ к ним через якоря. Сами ссылки на них расположить в начале страницы в виде содержания.
3. На главной странице сделать ссылки на две другие.
4. На обеих вновь добавленных страницах сделать ссылку возврата на главную страницу.

3. ОСНОВЫ СОЗДАНИЯ ПРЕЗЕНТАЦИЙ

Презентация (от лат. praesentatio) – общественное представление чего-либо нового, недавно появившегося, созданного, например: книги, журнала, кинофильма, телепрограммы, организации.

Презентация представляет собой сочетание компьютерной анимации, графики, видео, музыки и звукового ряда, которые организованы в единую среду. Как правило, презентация имеет сюжет, сценарий и структуру, организованную для удобного восприятия информации.

Отличительной особенностью презентации является ее интерактивность, то есть создаваемая для пользователя современными компьютерными средствами возможность взаимодействия с изображением.

Презентация – это обычно рекламный или информационный инструмент, позволяющий пользователю активно взаимодействовать с ним через меню управления.

На самом деле создание презентаций очень важная штука, прежде всего для компаний, которые задумываются о своем положении на рынке, конкуренции и дальнейшем развитии. Наряду с основными формами презентаций, такими как печатные формы (раздаточный материал в виде листовок, визиток ...), реклама на телевидении, сейчас получили широкое развитие презентации, выполненные в электронной форме (мультимедийные презентации). Они позволяют Вам выйти на так называемый «мультимедийный рынок», и Ваши возможности здесь безграничны (разместить на сайте, рассылать по электронной почте, использовать на конференциях, семинарах, создать электронный каталог продукции и этот список очень велик). Созданная презентация может быть выполнена в различных форматах (видах), в зависимости от источника воспроизведения:

Виды презентаций:

- презентация PowerPoint (имеют расширение PPT или PPTX для PowerPoint 2007 и выше) – самый простой вариант, который имеет очень много функций и возможностей;
- flash-презентация (расширение SWF либо скомпилированный в EXE, что позволяет запускать файл практически на любом компьютере) – флеш применим во многих сферах, его возможности практически безграничны. Создание презентации flash позволяет создать полноценную мультимедиа презентацию;
- видеопрезентации – их легко записать на носитель (флешку, DVD, CD, карту памяти). Созданные видео презентаций можно воспроизводить на любом бытовом плеере, компьютере либо проекторе.

Microsoft PowerPoint (полное название – Microsoft Office PowerPoint) – это программа для создания и проведения презентаций,

являющаяся частью Microsoft Office и доступная в редакциях для операционных систем Microsoft Windows и Mac OS.

Идея PowerPoint появилась у Боба Гаскина (Bob Gaskins), студента университета Беркли, который решил, что наступает век графических интерактивных материалов. В 1984 году Гаскинс присоединился к Forethought и нанял разработчика Денниса Остина (Dennis Austin). Боб и Деннис объединили усилия и создали программу Presenter. Деннис создал оригинальную версию программы с Томом Рудкиным (Tom Rudkin). Позже Боб решил сменить имя на PowerPoint, которое и стало названием конечного продукта.

В 1987 году вышел PowerPoint 1.0 для Apple Macintosh. Он работал в чёрно-белом цвете. Вскоре появились цветные Macintosh и новая версия PowerPoint не заставила себя ждать. Программное руководство с первой редакцией было уникально. Это была книга синего цвета в твердом переплете. Компания Forethought считала, что это лучше, чем куча исполняемых файлов помощи на компьютере и инструкций в электронном виде. Но обновление инструкции было необоснованно дорого. Идея с подобными книгами была вскоре заброшена.

Позже, в 1987 году, Forethought и её продукт были куплены Microsoft за 14 млн долларов. В 1990 году вышла версия для Windows. С 1990 года PowerPoint стал стандартом в наборе программ Microsoft Office.

В 2002 году вышла версия PowerPoint, которая не только была включена в пакет Microsoft Office XP, но также распространялась как отдельный продукт. В ней появились такие функции, как сравнение и смешение изменений в презентации, возможность задавать пути анимации для индивидуальных форм, создание пирамидальных, радиальных и целевых диаграмм, а также кругов Эйлера, панель задач для просмотра и выбора объектов буфера обмена, защита презентации паролем, автоматическая генерация фотоальбома, а также «умные теги» для быстрого выбора формата текста, скопированного в презентацию.

Microsoft PowerPoint 2003 не сильно отличается от предшественника. Он увеличивает эффективность работы в группе и теперь имеет возможность создания «Пакета для CD», который позволяет скопировать презентацию с мультимедийным содержанием и просмотреть её на компакт-диске.

Версия PowerPoint 2007 привносит основные изменения в интерфейс программы и увеличивает графические возможности.

Совместимость

PowerPoint является частью Microsoft Office. Это позволило PowerPoint стать наиболее распространённой во всем мире программой

для создания презентаций. Файлы презентаций PowerPoint часто пересылаются пользователями программы на другие компьютеры, что означает необходимую совместимость с ними программ конкурентов. Однако, поскольку PowerPoint имеет возможность подключения элементов других приложений через OLE, некоторые презентации становятся сильно привязанными к платформе Windows, что делает невозможным открытие данных файлов, например, в версии для Mac OS. Это привело к переходу на открытые стандарты, такие как PDF и OASIS OpenDocument.

Преобразование презентации PowerPoint в другой формат

Иногда приходится сталкиваться с такой проблемкой как конвертация презентации в графический формат (JPEG, GIF, PNG, BMP ...). Для этого необходимо: в главном меню программы выбрать пункт «Сохранить как», далее «Другие форматы», в поле «Тип файла» выбрать нужный и сохранить. Также таким способом можно конвертировать во многие другие форматы: PDF, XML, HTML, GIF, JPEG, PNG, EMF, RTF, ODP (Open Office). Презентация обычно содержит в себе текст, иллюстрации к нему и выдержана в едином графическом стиле.

Сегодня информационные технологии позволяют создавать презентации с использованием аудио- и видеовставок, делать презентации динамичными и интерактивными, использовать в них гипертекстовые ссылки.

Создание новой презентации

Существует два подхода к созданию презентации.

- 1) Создание презентаций с помощью шаблонов из встроенной библиотеки шаблонов PowerPoint.
- 2) Создание презентация с помощью конструктора.

PowerPoint содержит достаточное количество шаблонов презентаций, которые имеют профессионально разработанную структуру и содержание в различных предметных областях, а также грамотно оформленные, исходя из требований дизайнерского характера. Для создания презентации стандартной структуры и содержания имеет смысл воспользоваться представленными шаблонами. Это позволит сократить время на разработку, проектирование и составление презентации и, в то же время, даст возможность изменять структуру, содержание и дизайн будущей презентации с учётом особенностей поставленной задачи.

Вторым способом – создание презентации с помощью конструктора – следует воспользоваться, если подготавливаемая презентация нетипична и её структура, контекстное наполнение и дизайн детально разработаны.

Слайд – это отдельный кадр презентации, который может включать в себя заголовок, текст, графику, диаграммы, анимационные эффекты и т.д.

При создании презентации в PowerPoint можно использовать различные алгоритмы её построения. Однако единое для любого алгоритма то, что презентация состоит из слайдов, и для создания некоторой презентации сначала надо создать набор слайдов, которые впоследствии будут объединены в одну презентацию.

При проектировании слайда сначала можно выбрать его дизайн, структуру, а потом размещать на нём различные объекты (текст, диаграммы, рисунки, таблицы, схемы и др.), а можно сначала – структуру, разместить объекты, а потом дизайн.

Первый алгоритм более эффективен при построении слайдов, т.к. от дизайна зависят и расположение объектов, и их цветовая гамма, и оформление и др.

3.1. Практические задания по работе с PowerPoint

Задание 1. Моя визитка

Разработайте и создайте **презентацию «Моя визитная карточка»** (не менее 7 слайдов)

1 слайд: Моя визитка, личные данные, фото

2 слайд: Мои интересы, рассказ о хобби и интересах, используйте картинки, отображающие ваши интересы

3 слайд: Моё расписание (в табличном представлении)

4 слайд: Мои оценки (в виде диаграммы)

5 слайд: Мое генеалогическое древо (в виде организационной диаграммы - древовидной схемы)

6 слайд: Схема возраста моей семьи (в виде кольцевой или циклической или др. организационной диаграммы)

7 слайд: Моя анимация, мой фильм

8 слайд: Заключительный

Задание 2. Анимации

Создайте анимационные эффекты на объекты слайдов в ранее созданной презентации

1 слайд: Эффект **вылета**: Моя визитка, личные данные, фото

2 слайд: Эффект **вылета и улёта**: Мои интересы, рассказ о хобби и интересах, используйте картинки, отображающие ваши интересы

3 слайд: Эффект **качания и вращения**: Моё расписание (в табличном представлении)

4 слайд: Эффект **изменение пути перемещения – влево по спирали:** Мои оценки (в виде диаграммы)

5 слайд: Эффект **выделение - вращения:** Мое генеалогическое древо (в виде организационной диаграммы - древовидной схемы)

6 слайд: Эффект **мигания:** Схема возраста моей семьи (в виде кольцевой или циклической или др. организационной диаграммы)

7 слайд: Эффект **вылета, мигания и улета:** Моя анимация, мой фильм

8 слайд: Эффект на ваше усмотрение: Заключительный

Создайте анимационные эффекты на переход от слайда к слайду.

Все анимационные эффекты должны быть управляемые по времени.

Задание 3. Итоги

1. Создайте **Итоговый слайд** и создайте гиперссылки на каждый слайд, входящий в слайд-оглавление.

2. На каждом слайде организуйте навигацию возврата в **Итоговый слайд** с помощью графических кнопок.

3. Установите **Колонтитулы** текущей даты и номера страниц на слайдах.

4. Упакуйте презентацию для просмотра и показа на другом компьютере.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Боброва И.И. Дистанционный курс «Программные средства разработка web-страниц и презентаций»: электронный учебно-методический комплекс // Хроники объединенного фонда электронных ресурсов Наука и образование. Москва. – 2014. – Т. 1. № 12 (67). – С. 127.
2. Боброва И.И., Трофимов Е.Г. Практический курс. Информационные технологии в образовании. – 2-е изд., стереотип. – М.: Флинта, 2014. – 202 с.
3. Гаевский А.Ю., Романовский В.А. 100% самоучитель. Создание WEB-страниц и WEB-сайтов. – М.: Технолоджи-3000, 2008. – 464 с.
4. Ганиева Л.Ф., Аскарлова Н.А. Актуальные вопросы методики проведения лабораторных занятий по современным информационным технологиям для студентов психологов // Современная педагогика. – 2015. – № 11. – С. 20-24.
5. Информатика: учеб. пособие[электронный ресурс] / Е.Н. Гусева, И.Ю. Ефимова, Р.И. Коробков, К.В. Коробкова, И.Н. Мовчан, Л.А. Савельева. – 3-е изд., стереотип. – М.: Флинта, 2011. – 260 с. – Режим доступа: <http://www.knigafund.ru/books/116085>
6. Дакетт Д. Основы веб-программирования с использованием HTML, XHTML и CSS. – М.: Эксмо, 2010. – 768 с.
7. Климова Т.Е., Давлеткиреева Л.З., Романова М.В. Информационные и коммуникационные технологии в профессиональной подготовке студентов: учеб.-метод. пособие. – Магнитогорск: МаГУ, 2006. – 172 с.
8. Квинт И. HTML, XHTML и CSS на 100%. – СПб.: Питер, 2010. – 384 с.
9. Козн И. Полный справочник по HTML, CSS и JavaScript. – М.: Эком, 2007. – 1168 с.
10. Левин М. Самоучитель разработки Web-сайтов. HTML, CSS, графика, анимация, раскрутка (+CD). – М.: Триумф, 2008. – 400 с.
11. Хольцшлаг М. Языки HTML и CSS для создания Web-сайтов: учеб. пособие. – М.: Триумф, 2007. – 304 с.
12. Комолова Н. HTML. Самоучитель. – СПб.: Питер, 2008. – 268 с.
13. Подготовка текста выступления по защите диссертации: метод. пособие / сост. Т.Е. Климова, М.В. Романова. – Магнитогорск: МаГУ, 2009. – 341с.
14. Ташков П. Веб-мастеринг на 100 %: HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрутка. – СПб.: Питер, 2010. – 512 с.
15. Романов Е.П. Место преподавателя при подготовке и проведение компьютерных практикумов у студентов непрофильных направлений // Инновационная наука: международный научный журнал. – Уфа: АЭТЕРНА, 2015. – 304 с.

16. Романов Е.П. Лабораторный практикум по созданию Web-страниц. – Магнитогорск: МаГУ, 2010. – 71с.
17. Романова М.В. Савельев К.Н. Разработка электронных образовательных ресурсов // Новые информационные технологии в образовании: материалы VIII Международной научно-практической конференции. – Екатеринбург: Российский государственный профессионально-педагогический университет, 2015. – С. 364 – 367.
18. Фримен Э. Изучаем HTML, XHTML и CSS. – СПб.: Питер, 2010. – 656 с.

Учебное издание

Марина Викторовна РОМАНОВА
Елена Владимировна ЧЕРНОВА

РАЗРАБОТКА WEB-СТРАНИЦ И ПРЕЗЕНТАЦИЙ

Практикум

Издается полностью в авторской редакции

Подписано в печать 31.03.2017. Рег. № 37-17. Формат 60×84 ¹/₁₆. Бумага тип. № 1.
Плоская печать. Усл.печ.л. 4,50. Тираж 50 экз. Заказ 117.



Издательский центр ФГБОУ ВО «МГТУ им. Г.И. Носова»
455000, Магнитогорск, пр. Ленина, 38
Участок оперативной полиграфии ФГБОУ ВО «МГТУ им. Г.И. Носова»