

**Министерство образования и науки Кыргызской Республики
Кыргызский государственный технический университет
им. И. Раззакова**

Кафедра «Прикладная математика»

ИНФОРМАТИКА

Методические указания к выполнению практических работ по разделу
«Публикация Web документов»

Бишкек - 2010

РАССМОТРЕНО
На заседании кафедры «ПМ»
Прот. №5 от 05.02.2010г.

ОДОБРЕНО
Методическим советом ФИТ
Прот. №6 от 18.02.2010г.

УДК 681.3.01

Составитель - ст. преподаватель СТАМКУЛОВА Г.К., ДУШЕНОВА У.ДЖ.

Информатика: Методические указания к выполнению лабораторных работ по разделу «Публикация Web документов»

*/ КГТУ им. И. Раззакова. Сост: Стамкулова Г.К., Душенова У.Дж.
– Б.:ИЦ «Техник», 2010. – 35с.*

Представлены задания на разработку web-страниц, приведены общие указания к созданию web-страниц, задания для самостоятельной работы .

Предназначено для студентов по всем направлениям и всех форм обучения.

Рисунков: 12

Приложений: 1

Библиогр.: 4 наименов.

Рецензент доцент Уметалиев М.У.

Содержание

Введение	<u>4</u>
1. Лабораторная работа №1.	<u>6</u>
15 Указания к выполнению	7
2. Лабораторная работа №2. Применение языка HTML_	8
Указания к выполнению	9
3. Лабораторная работа №3. Гипертекстовые ссылки	<u>10</u>
Указания к выполнению	
4. Лабораторная работа №4Создание изображение и использование его на WEB	
странице	<u>12</u>
Указания к выполнению	<u>12</u>
5. Лабораторная работа №5. Создание таблицы.....	<u>13</u>
Указания к выполнению	<u>15</u>
Задание для самостоятельной работы	<u>16</u>
6. Лабораторная работа №6. Создание фреймов.....	<u>17</u>
Указания к выполнению	<u>18</u>
7. Лабораторная работа №7. Web-графка	<u>20</u>
Указания к выполнению	<u>22</u>
8. Лабораторная работа №. Работа в редакторе FrontPage Express.....	25
Указания к выполнению	27
Литература.....	<u>29</u>

Введение

Автономные Web-документы используют язык *HTML* (*HyperText Markup Language* — язык разметки гипертекста). *Гипертекст*, то есть *расширенный текст*, включает дополнительные элементы: иллюстрации, ссылки, вставные объекты. Под *разметкой* понимается использование специальных кодов, легко отделяемых от смыслового содержания документа и используемых для реализации гипертекста. Применение этих кодов подчиняется строгим правилам, определяемым спецификацией языка *HTML*.

Особенность описания документа средствами языка *HTML* связана с *принципиальной невозможностью достижения абсолютной точности воспроизведения исходного документа*. Предполагается, что документ будет широко доступен в Интернете, и поэтому *неизвестно, как будет организовано его воспроизведение*. Документ может быть представлен на графическом экране, выведен в чисто текстовом виде или просто «прочитан» программой синтеза речи. Разметка *HTML* во всех этих случаях должна быть принята во внимание. Поэтому язык *HTML* предназначен не для *форматирования* документа, а для его *функциональной разметки*.

Например, документы обычно начинаются с заголовков. Свойство части документа «быть заголовком» — это не особенность форматирования документа, а характеристика его содержания. Конкретное средство отображения документа (браузер) выбирает свой способ представления части документа, описанной как заголовок.

Управляющие конструкции языка *HTML* называются *тегами* и вставляются непосредственно в текст документа. Все теги заключаются в угловые скобки <...>

Сразу после открывающей скобки помещается ключевое слово, определяющее тег, например <DIV>. Теги *HTML* бывают *парными* и *непарными*. *Непарные* теги оказывают воздействие на весь документ или определяют разовый эффект в месте своего появления. При использовании парных тегов в документ добавляются *открывающий* и *закрывающий* теги, которые воздействуют на часть документа, заключенную между ними. Закрывающий тег отличается от открывающего наличием символа «/» (косая черта) перед ключевым словом (</DIV>). Закрытие парных тегов выполняется так, чтобы соблюдались правила вложения.

<I>На этот текст воздействуют два тега</I>

Эффект применения тега может видоизменяться путем добавления *атрибутов*. В парных тегах атрибуты добавляются *только к открывающему тегу*. Атрибуты представляют собой дополнительные ключевые слова, отделяемые от ключевого слова, определяющего тег, и от других атрибутов пробелами и размещаемые до завершающего тег символа «>». Способ применения некоторых атрибутов требует указания *значения атрибута*.

Значение атрибута отделяется от ключевого слова атрибута символом «=» (знак равенства) и заключается в кавычки.

```
<H1 ALIGN="LEFT">
```

Определение *HTML* как языка разметки основывается на том, что при удалении из документа всех тегов получается *текстовый* документ, совершенно эквивалентный по содержанию исходному *гипертекстовому* документу. Таким образом, при отображении документа *HTML* сами теги *не отображаются*, но влияют на способ отображения остальной части документа.

Если говорить о создании документов *HTML*, то можно представить себе два способа их формирования. Первый состоит в разметке существующего (или создаваемого) документа *вручную*. При этом автор или редактор добавляет в документ теги разметки. Эту работу можно выполнять в *текстовом редакторе* или *редакторе HTML*, имеющем специальные элементы управления для упрощения ввода тегов. В обоих этих случаях работа ведется средствами языка *HTML*, и человек, выполняющий эту работу, должен знать и уметь применять этот язык.

Принципы другого подхода можно понять на основе изучения работы текстовых процессоров. Информацию о форматировании документа также можно рассматривать как «разметку», добавляемую в формируемый документ. Однако для использования текстового процессора не требуется никаких знаний о формате документа и «языке разметки»: изменения, отображаемые на экране, вносятся в документ автоматически. Такой принцип соответствия экранного изображения реальному получил название *WYSIWYG* (от английского *What You See Is What You Get* — *Что видите, то и получаете*). Простейший редактор *WYSIWYG* для языка *HTML*, FrontPage Express, входит в состав операционной системы Windows 98. Первый способ позволяет создавать более универсальные, более качественные и более разнообразные документы. Второй способ проще освоить, так как он не требует знания языка *HTML*. Однако в этом случае используются средства *форматирования* вместо средств *описания*, что может иногда приводить к нежелательным последствиям. Процесс создания Web-документов сродни программированию и так же подвержен ошибкам. Независимо от того, каким способом создается документ, следует регулярно проверять его соответствие замыслу, просматривая его в различных браузерах. Для художественной оценки получающейся страницы следует обратиться к независимому мнению.

Лабораторная работа №1.

Интерактивные Web-Страницы

Web-страницы являются интерактивными по самой своей природе, связанной с использованием гиперссылок. Но это пассивная интерактивность, жестко заданная в рамках структуры Web-узла. Подлинная интерактивность, позволяющая получать от посетителей Web-страницы произвольные данные, достигается путем использования форм.

Форма на Web-странице представляет собой лишь набор *полей*, которые можно также рассматривать как *элементы управления* (рис. 19.7). Посетитель в процессе работы с Web-страницей заполняет форму, после чего отправляет ее. Далее поступившие данные обычно передают специальной программе, предназначенной для ее обработки (*программе или сценарию CGI*)

Внутри элемента формы располагаются поля формы. Они задаются при помощи различных тегов. Вот те, которые используются чаще всего.

- Непарный тег `<INPUT>` позволяет создавать различные элементы управления, в том числе текстовые поля и командные кнопки.

о Парный тег `<TEXTAREA>` определяет текстовые области.

в Парный тег `<SELECT>` позволяет создавать обычные и раскрывающиеся списки.

Отдельные пункты задаются при помощи парного тега `<OPTION>`, который допустим только внутри данного элемента (закрывающий тег `</OPTION>` можно опускать).

Тег `<INPUT>` должен содержать обязательный атрибут `TYPE=`, определяющий конкретный тип элемента управления. Вот основные возможные значения этого атрибута:

- "TEXT" — создается текстовое поле;
- » "PASSWD" — создается текстовое поле, но вводимая информация не отображается на экране («текстовое поле для ввода пароля»);
- в "CHECKBOX" — создается флажок, который может быть установлен или сброшен;
- "RADIO" — создается переключатель (из группы переключателей может быть включен только один);
- "SUBMIT" — создается *кнопка отправки* формы;
- в "IMAGE" — создается графическая кнопка отправки;
- в "RESET" — создается кнопка очистки формы, щелчок на которой возвращает форму к ее исходному состоянию;

Организация передачи данных, введенных в форму, осуществляется следующим образом. Каждый из тегов, задающих поля формы, должен иметь атрибуты NAME= и VALUE=. Атрибут NAME= определяет имя поля формы, атрибут VALUE= — значение поля. Для текстового поля и текстовой области атрибут VALUE= приобретает значение, соответствующее содержимому этого поля, заданному пользователем. Для элемента SELECT атрибут VALUE= соответствует содержимому выбранного элемента OPTION. Все переключатели одной группы должны иметь одинаковые значения атрибута NAME= и разные значения атрибута VALUE=.

По щелчку на кнопке отправки данные из формы передаются в виде пар текущих значений атрибутов NAME= и VALUE=, соединенных знаком равенства. Информация о флажке передается только в том случае, если он установлен. Если атрибуты NAME= и VALUE= определены для использованной кнопки отправки, соответствующие данные также передаются (это позволяет включать в форму несколько кнопок отправки). Данные, поступающие в таком виде, удобны как для ручной, так и для автоматической обработки.

Создавая формы, следует иметь в виду, что информация, передаваемую по электронной почте, нельзя считать конфиденциальной. Большинство браузеров может предупредить пользователя о возможности постороннего доступа к передаваемой информации. Эту особенность надо учитывать как при создании Web-страниц, содержащих формы, так и при заполнении форм на Web-страницах, встретившихся в Интернете.

Указания к выполнению : Создание простейшей Web-страницы

1. Запустите текстовый редактор Блокнот (Пуск ► Программы ► Стандартные ► Блокнот).

2. Введите следующий документ:

```
<HTML>
  <HEAD> <ТПП_Е>Заголовок документа<
    ДПП_Е>
  </HEAD>
  <BODY>
    Содержание
    Документа
  </BODY> </HTML>
```

3. Сохраните этот документ под именем first.htm.

Перед сохранением убедитесь, что сброшен флажок Не показывать расширения для зарегистрированных типов файлов (Пуск ► Настройка ► Свойства папки ► Вид). В противном случае редактор Блокнот может автоматически добавить в конец имени расширение .TXT.

4. Запустите программу Internet Explorer (Пуск ► Программы ► Internet Explorer).

5. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл first.htm.
6. Посмотрите, как отображается этот файл — простейший корректный документ *HTML*. Где отображается содержимое элемента TITLE? Где отображается содержимое элемента BODY?
7. Как отображаются слова «Содержание» и «документа», введенные в двух отдельных строчках? Почему? Проверьте, что происходит при уменьшении ширины окна.

Лабораторная работа №2

Применение языка HTML. Структура документа HTML

Все документы *HTML* имеют одну и ту же структуру, определяемую фиксированным набором *тегов структуры*. Документ *HTML* всегда должен начинаться с тега <HTML> и заканчиваться соответствующим закрывающим тегом (</HTML>). Внутри документа выделяются два основных раздела: *раздел заголовков* и *тело документа*, — идущие именно в таком порядке. Раздел заголовков содержит информацию, описывающую документ в целом, и ограничивается тегами <HEAD> и </HEAD>. В частности, раздел заголовков должен содержать общий заголовок документа, ограниченный парным тегом <TITLE>.

Основное содержание размещается в теле документа, которое ограничивается парным тегом <BODY>. Строго говоря, положение структурных тегов в документе нетрудно определить, даже если они опущены. Поэтому стандарт языка *HTML* *требует* только наличия тега <TITLE> (и, соответственно, </TITLE>). Тем не менее, при создании документа *HTML* опускать структурные теги не рекомендуется.

Простейший правильный документ *HTML*, содержащий все теги, определяющие структуру, может выглядеть следующим образом:

```
<HTML>  
<HEAD><TITLE>Заголовок документа<TITLE></HEAD>  
<BODY>  
Текст документа  
</BODY>  
</HTML>
```


Элементы HTML

Весь документ *HTML* можно рассматривать как «элемент *HTML*». Для непарных тегов элемент совпадает с тегом, который его определяет.

Большинство элементов языка *HTML* описывает части содержания документа и помещается между тегами `<BODY>` и `</BODY>`, то есть, внутри структурного элемента `BODY`. Такие элементы делят на *блочные* и *текстовые*. Блочные элементы относятся к частям текста уровня *абзаца*. Текстовые элементы описывают свойства *отдельных фраз* и еще более мелких частей текста.

Правила вложения элементов.

- *Элементы не должны пересекаться.* Другими словами, если открывающий тег располагается внутри элемента, то и соответствующий закрывающий тег должен располагаться внутри этого же элемента.
- *Блочные элементы могут содержать вложенные блочные и текстовые элементы.*
- *Текстовые элементы могут содержать вложенные текстовые элементы.*
- *Текстовые элементы не могут содержать вложенные блочные элементы.*

Все правила языка *HTML* можно рассматривать исключительно как «пожелания». Средство, используемое для отображения Web-документа, сделает все возможное, чтобы истолковать разметку наиболее разумным образом. Тем не менее, *гарантию* правильного воспроизведения документа дает только неукоснительное следование требованиям спецификации языка.

Указания к выполнению: Изучение приемов форматирования абзацев

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`, а его конкретное содержание может быть любым.
3. Введите заголовок первого уровня, заключив его между тегами `<H1 >` и `</H1 >`.
4. Введите заголовок второго уровня, заключив его между тегами `<H2>` и `</H2>`.
5. Введите отдельный абзац текста, начав его с тега `<P>`. Пробелы и символы перевода строки можно использовать внутри абзаца произвольно.
6. Введите тег горизонтальной линейки `<HR>`.
7. Введите еще один абзац текста, начав его с тега `<P>`.
8. Сохраните этот документ под именем `paragraph.htm`.
9. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
10. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте

файл paragraph.htm.

11. Посмотрите, как отображается этот файл. Установите соответствие между элементами кода *HTML* и фрагментами документа, отображаемыми на экране.
12. В этом упражнении мы создали документ *HTML* с разметкой абзацев. Мы определили, как влияют теги *HTML* на отображение соответствующих частей документа.

Лабораторная работа №3

Гипертекстовые ссылки.

Гипертекстовая ссылка является фрагментом текста документа и потому задается текстовым элементом, определяемым при помощи парного тега <A>. Этот элемент содержит обязательный атрибут, который не может быть опущен. В данном случае обязательным является атрибут H REF= (знак равенства показывает, что необходимо задать *значение* этого атрибута).

В качестве значения атрибута используется адрес *URL* документа, на который указывает ссылка. Она может указывать на произвольный документ, располагающийся на любом общедоступном узле сети (Web-узел, архив *FTP* и прочие). Например, открывающий тег ссылки может иметь вид <A HREF="<http://www.site.com/index.htm>">.

Адрес *URL* может быть задан в *абсолютной форме*, то есть начинаться с указания протокола и адреса Web-узла. Такая запись адреса используется, когда необходимо направить посетителя на другой Web-узел, и рассматривается как *внешняя ссылка*. При использовании *относительного адреса* в ссылке задается только относительный путь поиска для документа. В этом случае предполагается использование того же протокола и того же Web-узла, а ссылка рассматривается как *внутренняя*. Внутренняя ссылка сохраняет свою работоспособность в случае изменения адреса Web-узла как целого (например, в результате его переноса на другой сервер), поэтому при потенциальной возможности такого события следует отказываться от полного задания адресов в гиперссылках.

На Web-странице гипертекстовые ссылки выделяются цветом и подчеркиванием

Гиперссылки можно использовать для ссылки на мультимедийные файлы. Это удобно, так как в этом случае не приходится ждать загрузки мультимедийных файлов при работе с данной страницей. Если же требуется интегрирование объектов мультимедиа в Web-страницу, используют парный тег <OBJECT> или нестандартный непарный тег <EMBED>, который тоже поддерживается наиболее распространенными браузерами.

Полный формат гиперссылки включает возможность ссылки на *определенное место внутри страницы*. Но это можно сделать для страниц собственной разработки, пометив соответствующее место при помощи *якоря*. Якорь задается также при помощи парного тега <A>, но в роли обязательного выступает атрибут NAME=. Значение этого атрибута — произвольная последовательность латинских букв и цифр (пробелы недопустимы), рассматриваемая как *имя якоря*. Для ссылки на якорь его имя указывается в конце адреса URL после символа «#».

```
<AHREF="http://www.site.com/index.htm#adclress">
```

Указания к выполнению: Создание гиперссылок

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.
2. Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
3. Введите фразу: Текст до ссылки.
4. Введите тег: .
5. Введите фразу: Ссылка.
6. Введите закрывающий тег .
7. Введите фразу: Текст после ссылки.
8. Сохраните документ под именем link.htm.
9. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
10. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл link.htm.
11. Убедитесь в том, что текст между тегами <A> и выделен как ссылка (цветом и подчеркиванием).
12. Щелкните на ссылке и убедитесь, что при этом загружается документ, на который указывает ссылка.
13. Щелкните на кнопке Назад на панели инструментов, чтобы вернуться к предыдущей странице. Убедитесь, что ссылка теперь считается «просмотренной» и отображается другим цветом.

Лабораторная работа №4

Указания к выполнению: Создание изображения и использование его на Web-странице

1. Откройте программу Paint (Пуск ► Программы ► Стандартные ► Paint). Задайте размеры нового рисунка, например 50x50 точек (Рисунок ► Атрибуты).
2. Выберите красный цвет переднего плана и зеленый цвет фона. Залейте рисунок фоновым цветом.
3. Инструментом Кисть нанесите произвольный красный рисунок на зеленый фон.
4. Сохраните рисунок под именем pic1.gif (в формате GIF).
5. Дайте команду Рисунок ► Атрибуты. Установите флажок Использовать прозрачный цвет фона. Щелкните на кнопке Выбор цвета и выберите зеленый цвет, уже использованный на рисунке в качестве фонового.
6. Сохраните рисунок еще раз под именем pic2.gif (в формате GIF) и закройте программу Paint.
7. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.
8. Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
9. Введите произвольный текст (протяженностью 4-5 строк) и установите текстовый курсор в его начало.
10. Введите тег .
11. Сохраните документ под именем picture.htm.
12. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
13. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл picture.htm. Посмотрите на получившийся документ, обращая особое внимание на изображение.
14. Вернитесь в программу Блокнот и измените значение атрибута: ALIGN="TOP". Сохраните файл под тем же именем.
15. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
16. Вернитесь в программу Блокнот и измените значение атрибута: ALIGN="LEFT". Сохраните файл под тем же именем.
17. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
18. Вернитесь в программу Блокнот и добавьте в тег атрибуты: HSPACE=40 VSPACE=20. Сохраните файл под тем же именем.

19. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
20. Вернитесь в программу Блокнот и измените имя рисунка: SRC="pic2.gif". Сохраните файл под тем же именем.
21. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов. В чем различие между двумя созданными рисунками?

Лабораторная работа №5

Создание Таблицы

Таблица создается с помощью конструкции

```
<TABLE параметры >
<CAPTION>Заголовок таблицы</CAPTION>
  <TR параметры>
    <TH> ...
  </TH>
</TR>
<TR>
  <TD параметры>
</TD>
</TR>
</TABLE>
```

Тег <TABLE> ...</TABLE> - определяет начало и конец таблицы

имеет следующие параметры:

BACKGROUND="*.GIF|JPG" - определяет фоновое изображение таблицы

BGCOLOR = "цвет"/"#rrggb" - задает цвет фона таблицы

BORDER=n-задает толщину рамки таблицы в пикселах(n=0-рамки нет).

WIDTH, HEIGHT=n/n% - указывает размеры таблицы, если они должны быть жестко заданы.

ALIGN=RIGHT/LEFT/CENTER - определяет, как должна быть выровнена таблица относительно страницы.

CELLSPACING - число точек между отдельными ячейками в таблице.

CELLPADDING - число точек между рамкой и содержимым ячейки.

HSPACE, VSPACE=n - задает размер отступа по горизонтали и вертикали

FRAME = void(отсутствует)|above(вверху)|below (внизу) |hsides (горизонтальные) |vsides (вертикальные) |lhs (слева) |rhs (справа) |box (с 4-х сторон) |border (с 4-х сторон) - определяет, где браузер должен нарисовать границу вокруг таблицы.

SUMMARY = text - содержит краткое описание таблицы для использования с невизуальными браузерами

BORDERCOLOR = "цвет"/"#rrggb" - задает цвет главной границы таблицы

BORDERCOLORLIGHT = "цвет"/"#rrggb" - задает цвет светлой тени

BORDERCOLORDARK = "цвет"/"#rrggb" - задает цвет темной тени

тег <CAPTION параметры>...</CAPTION> - создает заголовок всей таблицы. И должен быть непосредственно после тега <TABLE>

Имеет параметр ALIGN=TOP/BOTTOM/LEFT/RIGHT/CENTER

VALIGN=TOP/BOTTOM

<TR параметры>...</TR> - задает в таблице строку ячеек.

параметры:

ALIGN=RIGHT/LEFT/CENTER -горизонтальное форматирование данных внутри ячеек

VALIGN - TOP/BOTTOM/MIDDLE/BASELINE - вертикальное форматирование данных внутри ячеек: могут иметь общую базовую линию (BASELINE).

BGCOLOR = "цвет"/"#rrggb" - задает цвет, используемый в строке тег <TH>...</TH> - нужен для задания заголовочных ячеек.

Он во всем совпадает с тегом <TD>, но в отличие от него, содержимое выдается жирным шрифтом и центрируется.

тег <TD параметры>...</TD> создает ячейки данных таблицы

параметры:

ALIGN=RIGHT/LEFT/CENTER -выравнивает содержимое ячейки

BACKGROUND="*.GIF|JPG" - задает фоновое изображение ячейки

BGCOLOR = "цвет"/"#rrggb" - задает цвет фона ячейки

HEIGHT=n/n% - задает высоту ячейки. Высота, заданная в первой ячейке будет использована для других ячеек строки

BORDERCOLOR = "цвет"/"#rrggb" - задает цвет границы ячейки

BORDERCOLORLIGHT = "цвет"/"#rrggb" - задает цвет светлой тени

BORDERCOLORDARK = "цвет"/"#rrggb" - задает цвет темной тени

WIDTH =n - задает ширину ячейки в пикселах.

COLSPAN=N - определяет, сколько колонок таблицы ячейка будет перекрывать.

ROWSPAN=N - определяет, сколько строк таблицы ячейка будет перекрывать.

NOWRAP - если этот параметр указан, то отменяется автоматическое обтекание текста для данной ячейки

Для управления выводом изображения в состав могут включаться такие атрибуты:

src – источник изображения, имя файла или путь к файлу, width – ширина

рисунка, height – высота рисунка, border – толщина рамки, alt – альтернативный

текст, который появляется при наведении мыши на рисунок.

HSPACE="X". Задает число свободных пикселей справа и слева от изображения.

VSPACE="X". Задает число свободных пикселей сверху и снизу от изображения.

Указания к выполнению

1. Необходимо создать таблицу, которая должна содержать все теги для создания таблиц.
2. Вставить любые изображения на Web-страницы.

Пример создания таблицы:

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.
2. Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>. В данном упражнении используется список номеров телефонов.
3. Введите тег <TABLE BORDER="10" WIDTH="100%">.
4. Введите строку: <CAPTION ALIGN="TOP">Список телефонов</CAPTION>.
5. Первая строка таблицы должна содержать заголовки столбцов. Определите ее следующим образом:

```
<TR BGCOLOR="YELLOW" ALIGN="CENTER">  
<TH>Фамилия<TH>Номер телефона
```
6. Определите последующие строки таблицы, предваряя каждую из них тегом <TR> и помещая содержимое каждой ячейки после тега <TD>
7. Последнюю строку таблицы задайте следующим образом:

```
<TRXTD ALIGN="CENTER" COLSPAN="2">На первом этаже  
здания имеется бесплатный телефон-автомат.
```
8. Завершите таблицу тегом </TABLE>.
9. Сохраните документ под именем table.htm.
10. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
11. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл table.htm.
12. Изучите, как созданная таблица отображается в программе Internet Explorer, обращая особое внимание на влияние заданных атрибутов.
13. Измените ширину окна обозревателя и установите, как при этом изменяется внешний вид таблицы.

ВЦ Мы познакомились с приемами создания таблиц средствами языка *HTML* для представления данных. Мы научились создавать таблицы и изменять их вид при помощи атрибутов тегов *HTML*.

Интерфейс общего примера:

Пример таблицы - Windows Internet Explorer

Файл Правка Вид Избранное Сервис Справка

DAEMON Tools


D:\Documents and Set

DAEMON Search

Пример таблицы

Пример создания таблицы

текст этой ячейки выровнен по центру	выравнивание содержимого вправо, текст внутри - зеленый
первая ячейка второй строки, вправо	вторая ячейка второй строки, влево



Задания для самостоятельной работы

1. В вышеуказанный пример добавить в каждую ячейку цвет.
2. Вставить в ячейку рисунок.
3. Создать таблицу расписания занятий на неделю.

Лабораторная работа №6. Фреймы.

Фреймы позволяют разделить окно браузера, отображаемое на экране монитора читателя, на отдельные области, содержимое каждой из которых может модифицироваться или перезагружаться независимо от других областей. Или наоборот, оставаться неизменным, в то время как содержимое других областей обновляется.

Тег <FRAME> - определяет один фрейм в составе тега <FRAMESET>

Параметры:

BORDERCOLOR = "цвет"/"#rrggbb"

FRAMEBORDER = 1/0 (yes|no)-задает вывод трехмерного разделителя м/у текущим и окружающими его фреймами. (1-появление границы)

MARGINWIDTH = n - задает кол-во пикселей в промежутке м/у левым и правым краем фрейма и его содержимым

MARGINHEIGHT= n - задает кол-во пикселей в промежутке между верхней и нижней границей фрейма и его содержимым
NORESIZE- предотвращает изменение размера фрейма пользователем
NAME = text - назначает имя фрейму и могут использоваться в ссылках
SCROLLING = yes|no|auto
SRC= url - задает местонахождение исходного HTML-файла, который должен появиться во фрейме.

Тег <FRAMESET> ... </FRAMESET> определяет образующее группы фреймов множество фреймов или др. групп фреймов

Параметры:

BORDER = n - задает толщину границ фрейма (в пикселях)

BORDERCOLOR = "цвет"/"#rrggbb"

COLS = список - устанавливает количество и размер столбцов в группе фреймов

FRAMEBORDER = 1|0 - определяет вывод 3-мерных разделителей м/у фреймами

FRAMESPACING = N (MSIE) - добавляет дополнительный свободный интервал м/у фреймами

ROWS = список (n| n% |*) - задает кол-во и размер строк в группе фреймов.

Кол-во строк определяется по кол-ву значений в списке

Тег <IFRAME> ... </IFRAME> определяет в документе плавающий фрейм

Параметры:

ALIGN=TOP/BOTTOM/MIDDLE/LEFT/RIGHT/CENTER - задает выравнивание на странице в потоке текставнутреннего фрейма

FRAMEBORDER = 1|0 - включает или отключает изображение объемных границ фрейма

WIDTH, HEIGHT=n|n% - задает ширину и высоту фрейма

HSPACE, VSPACE=n - определяет интервал влево, вправо и сверху, снизу от фрейма

MARGINWIDTH, MARGINHEIGHT=n - задает размер интервала между краями фрейма и его содержимым

NAME=имя - назначает фрейму имя, по которому будет осуществляться ссылка.

NORESIZE - предотвращает изменение размера фрейма со стороны пользователя

SCROLLING=yes|no|auto- определяет полосы прокрутки

SRC=URL - URL HTML-документа, который первым отобразится во фрейме

В настоящее время фреймы поддерживаются тремя основными браузерами Сети – Internet Explorer, Netscape и Opera. Чтобы владельцы других браузеров не оказались в недоумении, увидев перед собой пустую страницу, следует помещать в HTML-документ, содержащий тэг <FRAMESET>, специальный тэг <NOFRAMES> с пояснительным сообщением, например:

<NOFRAMES>

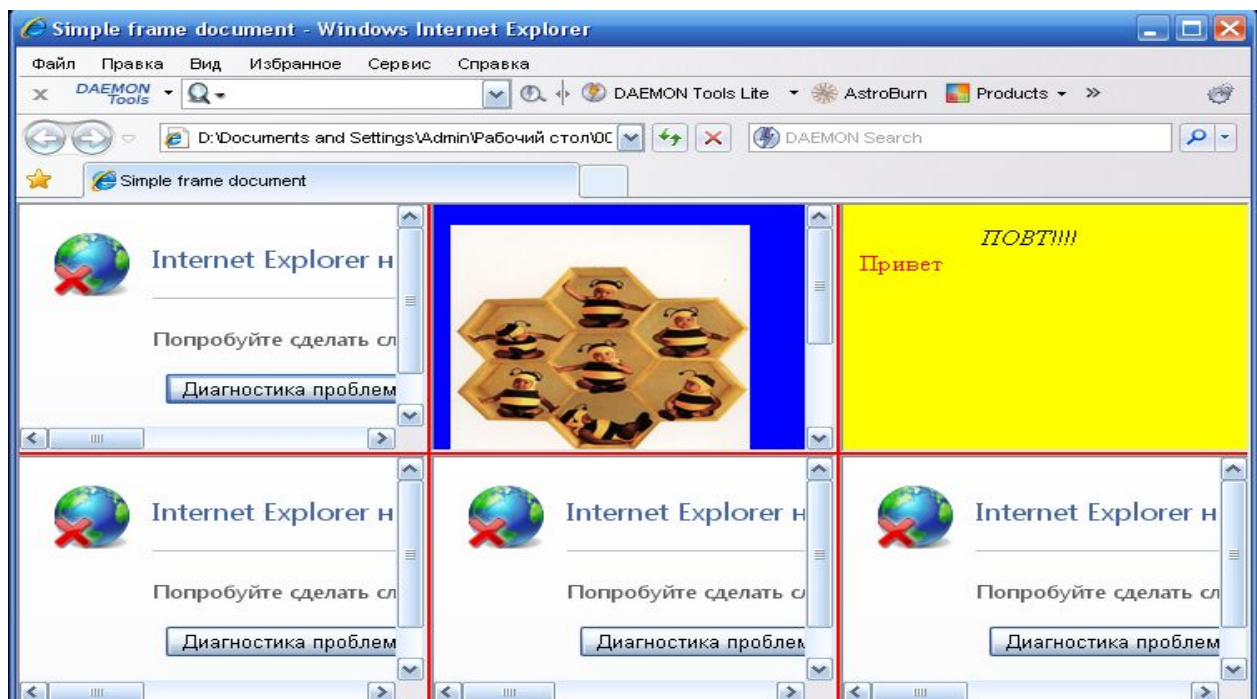
На этой странице используются фреймы; Ваш браузер их не поддерживает
</NOFRAMES>.

Указания к выполнению:

1. Запустите текстовый редактор Блокнот (Пуск ► Программы ► Стандартные ► Блокнот).
2. Введите следующий документ:

```
<HTML> <HEAD>
  <TITLE>Описание фреймов</TITLE> </HEAD>
<FRAMESET ROWS="60%,*">
  <FRAME SRC="table.htm">
<FRAMESET COLS="35%,65%" NORESIZE>
<FRAME SRC="first.htm"> <FRAME
SRC="lnks.htm"> </FRAMESET> </HTML>
```
3. Сохраните этот документ под именем frames.htm.
4. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
5. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл frames.htm.
6. Изучите представление нескольких созданных ранее документов, в отдельных фреймах.
7. Посмотрите, что происходит при изменении ширины окна обозревателя.
8. Проверьте, можно ли изменить положение границ фреймов методом перетаскивания при помощи мыши.
9. Щелкните на ссылке, имеющейся в одном из фреймов, и посмотрите, как будет отображен новый документ.
10. Щелкните на кнопке Назад на панели инструментов и убедитесь, что возврат к предыдущему документу нарушает структуру фреймов.
11. Вернитесь в программу Блокнот и измените структуру и параметры фреймов по своему усмотрению. Сохраните документ под тем же именем.
12. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Убедитесь, что измененный вид Web-страницы соответствует замыслу. Если это не так, вернитесь в программу Блокнот, найдите и исправьте ошибки.

Мы научились отображать в рамках одной Web-страницы несколько документов при помощи фреймов. Мы узнали, как фреймы отображаются обозревателем Internet Explorer. Мы исследовали особенности навигации с помощью гиперссылок по Web-странице, содержащей фреймы.



Лабораторная работа №7.

Web-графика

Сегодня графические элементы Web-страниц используют два основных формата — GIF и JPEG (новый формат PNG пока еще нельзя считать общепринятым). Все графические браузеры, предназначенные для отображения Web-страниц на экране компьютера, способны распознавать и отображать файлы этих форматов. Для вставки рисунка используется текстовый элемент, задаваемый непарным тегом . Тег должен содержать обязательный атрибут SRC=, задающий адрес URL файла с изображением в относительной или абсолютной форме.

```
<IMG SRC=" Трутни . jpg ">
```

При отображении рисунка браузер по умолчанию использует его реальные размеры. Если рисунок необходимо отмасштабировать, применяют атрибуты WIDTH= и HEIGHT=, задающие ширину и высоту рисунка (в пикселях), соответственно.

```
<IMG SRC=Трутни.jpg " WIDTH="180" HEIGHT="200">
```

Внешний вид Web-страницы зависит от того, как именно рисунок располагается на ней. Так как рисунок задается как текстовый элемент, находящийся внутри какого-то абзаца, по умолчанию он рассматривается как встроенное изображение,

включаемое в строку текста. Чтобы изображение отображалось автономно, его включают в отдельный абзац.

Для изображения, которое действительно включено в строку, можно задать режим взаимодействия с текстом с помощью атрибута

ALIGN=.

```
<IMG SRC="picture3. jpg " ALIGN="BOTTOM">
```

Этот атрибут может принимать три значения:

1. ALIGN="BOTTOM" - нижняя граница изображения совмещается с основанием текстовой строки;
2. ALIGN="MIDDLE" - середина изображения совмещается с серединой текстовой строки;
3. ALIGN="TOP" - верхняя граница изображения выравнивается по верхнему обрезу текстовой строки.

Однако более предпочтительно использование «плавающего» изображения, обтекаемого текстом, что также достигается использованием атрибута ALIGN=:

ALIGN="LEFT" - изображение размещается у левого края страницы, а последующий текст размещается справа от него;

ALIGN="RIGHT" - изображение размещается у правого края страницы, а последующий текст размещается слева от него.

В этом случае рекомендуется помещать тег в самое начало соответствующего абзаца.

Однако нормальный режим обтекания требует, чтобы между текстом и изображением оставался некоторый промежуток: *HSPACE* - горизонтальный отступ от графического изображения. *VSPACE* - вертикальный отступ.

Так как изображение задается как текстовый элемент, оно может быть помещено внутрь другого текстового элемента, например, задающего гиперссылку (тег <A>). В этом случае изображение становится изображением-ссылкой. При отображении документа на экране компьютера такое изображение отличается синей рамкой и изменением формы указателя при наведении.

Еще один способ применения это фоновый рисунок, на компьютере рабочая область окна заполняется этим рисунком последовательно (как паркетом), считая от верхнего левого угла документа или окна. Фоновый рисунок задается с помощью атрибута BACKGROUND= в теге <BODY>. Значением этого атрибута должен быть абсолютный или относительный адрес URL для файла с изображением.

```
<BODY BACKGROUND="waves.gif" TEXT="YELLOW">
```

Форматирование текста

<I> ... </I> или ... -курсив

... или ... - жирным шрифтом для всего документа можно задать с помощью тега .<U> (подчеркнутый текст) <S> (вычеркнутый текст).

<BASEFONT SIZE=n> - Базовый размер шрифта n= от 1 до 7

... - Изменяет размеры и цвет шрифтов

Параметры:

SIZE=от 1 до 7 или относительно базового -2 или +1

COLOR - указывает цвет для текста.

FACE - расширение MSIE и Netscape Navigator. Позволяет задать гарнитуру (или список гарнитур, из имеющихся в системе шрифтов будет выбран наиболее подходящий) для текста.

Пример:

Список

Язык HTML рассматривает символы конца строки и пробелы особым образом. Любая последовательность, состоящая только из пробелов и символов конца строки, при отображении документа рассматривается как одиночный пробел. Это, в частности, означает, что символ конца строки даже не осуществляет перехода на новую строку (для этой цели используется текстовый элемент, задаваемый непарным тегом
).

В качестве ограничителя абзацев может также использоваться горизонтальная линейка. Этот элемент задается непарным тегом <HR>. При отображении документа на экране линейка разделяет части текста друг от друга. Ее длина и толщина задаются атрибутами тега <HR>.

<HRAUGN="RIGHT"SIZE="10"WIDTH="50%">

Этот тег создает горизонтальную линейку шириной в 10 пикселей, занимающую половину ширины окна и расположенную справа.

Цвет	Код
Black (черный)	#000000
Maroon {темно-бордовый)	#800000
Green (зеленый)	#008000
Olive (оливковый)	#808000
Navy (темно-синий)	#000080
Purple (фиолетовый)	#800080
Teal (чирок)	#008080
Gray (серый)	#808080
Silver (серебряный)	#C0C0C0

Red (красный)	#FF0000
Lime (известь)	#00FF00
Yellow (желтый)	#FFFF00
Blue (синий)	#0000FF
Fuchsia (фуксия)	#FF00FF
Aqua (аква)	#00FFFF
White (белый)	#FFFFFF

7.1. Создание изображения и использование его на Web-странице

1. Откройте программу Paint (Пуск ► Программы ► Стандартные ► Paint). Задайте размеры нового рисунка, например 50x50 точек (Рисунок ► Атрибуты).
2. Выберите красный цвет переднего плана и зеленый цвет фона. Залейте рисунок фоновым цветом.
3. Инструментом Кисть нанесите произвольный красный рисунок на зеленый фон.
4. Сохраните рисунок под именем pic1.gif (в формате GIF).
5. Дайте команду Рисунок ► Атрибуты. Установите флажок Использовать прозрачный цвет фона. Щелкните на кнопке Выбор цвета и выберите зеленый цвет, уже использованный на рисунке в качестве фонового.
6. Сохраните рисунок еще раз под именем pic2.gif (в формате GIF) и закройте программу Paint.
7. Если это упражнение выполняется не сразу после предыдущего, откройте документ first.htm в программе Блокнот.
8. Удалите весь текст, находящийся между тегами <BODY> и </BODY>. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега <BODY>.
9. Введите произвольный текст (протяженностью 4-5 строк) и установите текстовый курсор в его начало.
10. Введите тег .
11. Сохраните документ под именем picture.htm.
12. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
13. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл picture.htm. Посмотрите на получившийся документ, обращая особое внимание на изображение.
14. Вернитесь в программу Блокнот и измените значение атрибута: ALIGN="TOP". Сохраните файл под тем же именем.
15. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели

- инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
16. Вернитесь в программу Блокнот и измените значение атрибута: `ALIGN="LEFT"`. Сохраните файл под тем же именем.
 17. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
 18. Вернитесь в программу Блокнот и добавьте в тег `` атрибуты: `HSPACE=40` `VSPACE=20`. Сохраните файл под тем же именем.
 19. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов.
 20. Вернитесь в программу Блокнот и измените имя рисунка: `SRC="pic2.gif"`. Сохраните файл под тем же именем.
 21. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы при изменении атрибутов. В чем различие между двумя созданными рисунками?

7.2. Приемы форматирования текста

1. Если это упражнение выполняется не сразу после предыдущего, откройте документ `first.htm` в программе Блокнот.
2. Удалите весь текст, находящийся между тегами `<BODY>` и `</BODY>`. Текст, который будет вводиться в последующих пунктах этого упражнения, необходимо поместить после тега `<BODY>`, а его конкретное содержание может быть любым.
3. Введите тег `<BASEFONT SIZE="5" COLOR="BROWN">`. Он задает вывод текста по умолчанию увеличенным шрифтом и коричневым цветом.
4. Введите произвольный абзац текста, который будет выводиться шрифтом, заданным по умолчанию. Начните этот абзац с тега `<P>`.
5. Введите теги: `<P> `.
6. Введите очередной абзац текста, закончив его тегом ``.
7. В следующем абзаце используйте по своему усмотрению парные теги: `` (полужирный шрифт), `<I>` (курсив), `<U>` (подчеркивание), `<S>` (вычеркивание), `<SUB>` (нижний индекс), `<SUP>` (верхний индекс).
8. В следующем абзаце используйте по своему усмотрению парные тега: `` (выделение), `` (сильное выделение), `<CODE>` (текст программы), `<KBD>` (кла-

- виатурный ввод), <SAMP> (пример вывода), <VAR> (компьютерная переменная).
9. Сохраните полученный документ под именем format.htm.
 10. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
 11. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл format.htm.
 12. Изучите, как использованные элементы *HTML* влияют на способ отображения текста.
 13. Вернитесь в программу Блокнот и измените документ так, чтобы элементы, задающие форматирование, были вложены друг в друга. Сохраните документ под тем же именем.
 14. Вернитесь в программу Internet Explorer и щелкните на кнопке Обновить на панели инструментов. Посмотрите, как изменился вид страницы.

Лабораторная работа №8

Работа в редакторе FrontPage Express

Начиная работу в редакторе FrontPage Express, следует отдавать себе отчет, что в результате его применения получается документ *HTML*, построенный по тем же правилам, что и создаваемый вручную. Этот факт сразу же определяет возможности и ограничения в работе этого редактора.

- Все функции редактора FrontPage Express однозначно реализуются тегами *HTML*.
- Редактор FrontPage Express не имеет средств, которые нельзя было бы представить в виде тегов *HTML*.
- Пользователь обычно не знает, какие именно средства *HTML* используются для достижения заданного эффекта и насколько корректно они применяются.

Редактор FrontPage Express «ориентирован» на применение обозревателя Internet Explorer, так что создаваемый им код *HTML* наиболее адекватно отображается именно в этом браузере. В частности, FrontPage Express позволяет использовать «бегущую строку», средство, которое не входит в стандарт *HTML*, но поддерживается Internet Explorer. Из числа стандартных средств *HTML* редактор FrontPage Express не поддерживает фреймы (точнее говоря, создание документов описания фреймов).

Создание и редактирование документа

Окно программы FrontPage Express представляет собой комбинацию *окна редактора* и *окна браузера*. Документ *HTML* отображается редактором, как специфическим браузером, отображающим даже обычно невидимые элементы (такие как якоря). В то же время, этот текст можно редактировать средствами, аналогичными имеющимся в текстовом процессоре.

В качестве основного средства форматирования используется Панель инструментов форматирования. Она содержит:

- раскрывающийся список Изменение стиля, позволяющий выбрать стиль оформления абзаца (соответствующий стандартным функциональным элементам *HTML*);
- раскрывающийся список Изменение шрифта, позволяющий выбрать гарнитуру шрифта (наличие такого же шрифта в ходе просмотра документа через Интернет не гарантируется);
- кнопки увеличения и уменьшения размера текста (в относительных единицах *HTML*);
- кнопки выбора начертания;
- кнопки выбора выравнивания текста (с помощью атрибута `AUGN=` в теге абзаца `<P>`);
- в кнопки создания маркированных (неупорядоченных) и нумерованных (упорядоченных) списков;
- кнопки задания отступа текста (на основе некорректного использования элементов *HTML*).

Дополнительные элементы форматирования, не вынесенные на панель инструментов, задаются в отдельных диалоговых окнах. Их можно открыть, например, с помощью команд **Формат ► Шрифт** и **Формат ► Список**.

Для создания гиперссылки надо выделить фрагмент текста, который будет использоваться как ссылка, и дать команду **Вставка ► Гиперссылка**. В открывшемся диалоговом окне **Создать гиперссылку** выбирают вкладку **WWW** для создания внешней ссылки, вкладку **Открыть страницы** для создания ссылки на другую страницу своего узла (используется относительный адрес) и вкладку **Новая страница** для одновременного создания ссылки и соответствующего документа, который немедленно открывается для редактирования.

Поле **Кадр** назначения служит для указания имени фрейма, в котором будет открываться страница, а поле **Закладка** — для указания якоря, который при этом используется.

Созданная гиперссылка отображается в окне программы FrontPage Express так же, как и в окне броузера: синим цветом и с подчеркиванием. Чтобы проверить работоспособность ссылки, следует щелкнуть на ней правой кнопкой мыши и выбрать в контекстном меню команду Перейти по ссылке.

Таблицы и формы невозможно создать только форматированием. Чтобы создать таблицу, используют команду Таблица ► Вставить таблицу. В открывшемся диалоговом окне указывают размер таблицы (в ячейках), а также дополнительные параметры, реализуемые как атрибуты соответствующих тегов. Чтобы занести информацию в таблицу, следует установить курсор в нужную ячейку и начать ввод. При работе с ячейками таблицы можно применять любые команды форматирования.

Специальной команды создания формы редактор FrontPage Express не имеет. Однако при добавлении *поля формы* (через меню Вставка ► Поле формы) создается и форма, включающая это поле. При добавлении последующих полей следует следить за тем, чтобы они включались в ту же самую форму.

В языке *HTML* свойства элемента задаются атрибутами тега. Редактор FrontPage Express обеспечивает их задание с помощью специальных диалоговых окон. Чтобы открыть такое диалоговое окно, следует щелкнуть на редактируемом элементе правой кнопкой мыши. В нижней части контекстного меню располагаются команды, относящиеся к элементам документа, рассматриваемым редактором как открытые. Порядок следования команд соответствует порядку вложения элементов. Выбор одной из этих команд приводит к открытию диалогового окна свойств соответствующего элемента. Элементы управления в этом диалоговом окне соответствуют атрибутам открывающего тега для выбранного элемента.

Дополнительные объекты вставляются в редактируемый документ при помощи меню Вставка. Например, для вставки изображения используется команда Вставка ► Изображение. Если в диалоговом окне Изображение щелкнуть на кнопке Обзор, можно выбрать файл с изображением. В ходе создания документа различные графические форматы автоматически преобразуются в форматы *GIF* и *JPEG*. Сохранение преобразованного файла осуществляется при сохранении документа.

Другие элементы, которые можно разместить на странице, — это:

- горизонтальная линейка (Вставка ► Горизонтальная линия);
- видеозапись (Вставка ► Видеозапись);
- фоновое звуковое сопровождение (Вставка ► Фоновый звук);
- «бегущая строка» (Вставка ► Бегущая строка, этот элемент воспроизводится только обозревателем Internet Explorer).

Ограниченность возможностей редактирования при помощи FrontPage Express иногда приводит к необходимости вмешательства непосредственно в создаваемый код *HTML*. Это можно осуществить двумя способами. Во-первых, можно дать команду л Вставка ► Разметка HTML В этом случае открывается диалоговое окно Разметка л HTML, в котором можно вводить произвольный текст с разметкой. Редактор FrontPage

Указания к выполнению:

Создание Web-документа с помощью редактора FrontPage Express

1. Запустите программу FrontPage Express (Пуск ► Программы ► Стандартные ► Средства Интернета ► FrontPage Express).
2. Введите в программе FrontPage Express произвольный текст документа.
3. С помощью панели инструментов форматирования отформатируйте текст по собственному усмотрению.
4. Для создания таблицы щелкните на кнопке Вставить таблицу на стандартной панели инструментов.
5. Для добавления иллюстраций используйте кнопку Вставить изображение на стандартной панели инструментов. Иллюстрации возьмите из папки C:\Windows.
6. Дайте команду Файл ► Сохранить, щелкните на кнопке Как файл и задайте имя файла wysiwyg.htm. Подтвердите сохранение изображений, требующих преобразования формата.
7. Запустите обозреватель Internet Explorer (Пуск ► Программы ► Internet Explorer).
8. Дайте команду Файл ► Открыть. Щелкните на кнопке Обзор и откройте файл wysiwyg.htm.
9. Убедитесь, что созданный документ правильно отображается обозревателем. Обратите внимание на наличие отличий вида документа при отображении в обозревателе и в программе FrontPage Express.
10. Измените ширину окна обозревателя и посмотрите, как при этом меняется вид документа.
11. Вернитесь в программу FrontPage Express и дайте команду Вид ► HTML.
12. Изучите автоматически сгенерированный код *HTML*, определите, как с помощью тегов *HTML* реализованы использованные команды форматирования
13. Оцените качество получившегося кода *HTML*.

Литература

1. Кэмпбел Марк «Строим Web сайт» Москва 2006г.
2. Основы Web-дизайна. Вспомогательное руководство. Москва-Санкт-Петербург-Киев, Ксения Гиз, Александра Холмз,2002
3. Самоучитель создания Web-страниц и Web-сайтов /Под редакцией Комяшина В.Б., - М. «Издательство Тримиф», 2001