

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ УКРАИНЫ**  
**ХАРЬКОВСКИЙ НАЦИОНАЛЬНЫЙ УНИВЕРСИТЕТ**  
**ГОРОДСКОГО ХОЗЯЙСТВА имени А. Н. БЕКЕТОВА**

**МЕТОДИЧЕСКИЕ УКАЗАНИЯ**

для выполнения лабораторных, практических,  
самостоятельных и контрольных работ  
по курсу

**ИНФОРМАТИКА**

**Основы Web-дизайна**

*(для студентов 1-го курса дневной и заочной форм обучения  
образовательно-квалификационного уровня бакалавр,  
направлений подготовки 6.030504 – Экономика предприятия  
и 6.030509 – Учет и аудит)*

Методические указания для выполнения лабораторных, практических, самостоятельных и контрольных работ по курсу «Информатика» (для студентов 1-го курса дневной и заочной форм обучения образовательно-квалификационного уровня бакалавр, направлений подготовки 6.030504 – Экономика предприятия и 6.030509 – Учет и аудит) / Харьков. нац. ун-т гор. хоз-ва им. А. Н. Бекетова; сост.: Б. И. Погребняк. О. М. Штельма. – Харьков : ХНУГХ им. А. Н. Бекетова, 2015. – 60 с.

Составитель: Б. И. Погребняк  
О. М. Штельма

Рецензент: к.ф.-м.н., доц. А. Б. Костенко

Рекомендовано кафедрой «Прикладной математики и информационных технологий», протокол № 1 от 30 августа 2014 г.

## **Оглавление**

Введение .....	4
Содержание контрольной работы.....	5
Выбор варианта .....	5
Порядок оформления .....	6
Лабораторная работа № 1 Создание и редактирование Web-страниц.....	7
Лабораторная работа № 2 Форматирование Web-страниц .....	22
Лабораторная работа № 3 Гиперссылки и графика .....	34
Лабораторная работа № 4 Таблицы.....	47
Список источников.....	59

# Введение

## Структура лабораторной работы

Каждая лабораторная работа состоит из таких частей:

1. Цель работы,
2. Краткие теоретические сведения,
3. Ход работы,
4. Индивидуальные задания и
5. Контрольные вопросы.

Ход работы состоит из отдельных Шагов, которые, в свою очередь, состоят из отдельных Пунктов. В Пунктах описаны требуемые действия и получаемые (в случае правильного выполнения) результаты.

## Порядок выполнения

Лабораторные работы сконструированы так, что максимальный обучающий эффект, при минимальной затрате сил, достигается если их выполнять следующим образом:

1. Лабораторные работы необходимо выполнять **последовательно** – в том порядке, в котором они здесь приведены.
2. **Обязательно внимательно прочитать** Краткие теоретические сведения.
3. Ход работы выполнять **строго** Шаг за Шагом, и Пункт за Пунктом, **без пропусков**.
4. Имена файлов, каталогов тегов и других объектов **вводить так**, как указано в тексте. В противном случае могут быть нарушены связи, что приведет ошибке выполнения, которая, в свою очередь, может обнаружиться через несколько Пунктов, или даже Шагов.
5. Индивидуальные задания выполнять **только после полного завершения** Хода работы. Если при выполнении Индивидуального задания возникают вопросы, то необходимо вернуться к Кратким теоретическим сведениям или Ходу работы – если предыдущая часть Лабораторной работы была освоена полностью, то Индивидуальные задания вопросов вызывать не должны.
6. Если все предыдущие пункты выполнены правильно, то ответы на Контрольные вопросы не вызовут никаких затруднений.
7. Теперь Лабораторная работа готова к защите у преподавателя, которая состоит в подтверждении полученных знаний по указанной теме.

## Обработка ошибок

Если в ходе выполнения очередного Пункта получен результат, несоответствующий требуемому, то последовательность действий должна быть такой:

1. Выполнить более внимательно текущий пункт еще раз.
2. Если результат снова не соответствует ожидаемому, то необходимо повторить весь текущий Шаг еще раз (для этого лабораторные работы и были разбиты на Шаги).
3. Если и в этом случае ошибка не была устранена, то Лабораторную работу надо начать выполнять с самого начала – с **внимательного прочтения** Кратких теоретических сведений.

## Содержание контрольной работы

В Контрольной работе необходимо привести выполнение Индивидуальных заданий по Лабораторным работам в соответствии со своим вариантом, и оформить их соответствующим образом. Каждая Лабораторная работа – это одна из тем освоения материала. Лабораторные работы выполняются на компьютере, а правильность их выполнения контролируется по приводимым в тексте результатам. Если полученный в ходе выполнения Лабораторной работы результат не соответствует приводимому в тексте, то необходимо вернуться на один, или несколько шагов назад, и более внимательно выполнить указанные действия. И только после правильного выполнения всех Шагов и Пунктов можно приступать к выполнению Индивидуального задания. Конкретное содержание отчета по каждой Лабораторной работе приведено в ее Индивидуальном задании.

## Выбор варианта

Вариант Индивидуального задания формируется в соответствии с:

- указанием преподавателя, или
- номером по списку группы, или
- по 2-м последним цифрам номера зачетной книжки.

По 2-м последним цифрам номера зачетной книжки вариант работы вычисляется как **остаток от целочисленного деления 2-х последних цифр номера зачетной книжки на 30** либо **последовательным вычитанием из 2-х последних цифр номера зачетной книжки по 30 до тех пор, пока остаток не**

**будет меньше 30.** Например, две последние цифры номера зачетной книжки «07», тогда  $7 \% 30 = 7$  т. е. номер варианта индивидуального задания будет равен «7»; если две последние цифры номера зачетной книжки будут, например, «68», тогда номер варианта будет:  $68 \% 30 = 8$  или  $68 - 30 = 38 - 30 = 8$ , т.е. «8».

## Порядок оформления

Работа оформляется на отдельных листах формата А4 (210мм × 297мм) либо в ученической тетради. Текст работы может быть набран шрифтом Times New Roman 14 пунктов через 1,5 интервала или написан от руки. На титульной странице обязательно необходимо указать название дисциплины и свой вариант.

При оформлении Индивидуальных заданий необходимо привести:

1. Номер и название Лабораторной работы.
2. Конкретные результаты выполнения Индивидуального задания, и как они были получены (например, HTML-коды создаваемых Web-страниц и соответствующие им эскизы, и т. д.). Более конкретное содержание отчета по каждой Лабораторной работе приведено в тексте ее Индивидуального задания.

Для дневной формы обучения необходимо также письменно ответить на Контрольные вопросы, приведенные в конце каждой Лабораторной работы, в форме «вопрос – ответ».

Настоятельно рекомендуется также в тетради для выполнения Лабораторных работ делать записи наиболее существенных (трудных, непонятных, интересных и т. п.) моментов, возникающих по ходу их выполнения.

# Лабораторная работа № 1

## Создание и редактирование Web-страниц

### *Цель работы*

Освоение и приобретение практических навыков работы с инструментарием для создания Web-страниц, их создание и редактирование, управление цветом текста и фона.

### *Краткие теоретические сведения*

Для создания Web-страниц используются различные программные средства, продуктивная и комфортная работа с которыми в большой степени зависит от их оптимальной совместной настройки. В простейшем случае HTML-документ можно подготовить при помощи стандартного текстового редактора, который по умолчанию имеется в любой операционной системе. Например, в Windows – это Блокнот (Notepad), а в Linux – Mousepad. Посмотреть же созданную Web-страницу можно в программе-браузере, например, Firefox – для Linux или Internet Explorer – для Windows. Поскольку текстовые редакторы изначально предназначены для подготовки текстовых документов, то по умолчанию они при сохранении документа либо вообще не ставят ни какого расширения имени файла (например, Mousepad в Linux), либо используют расширение `.txt` (Блокнот в Windows). Поэтому при сохранении Web-страниц необходимо явно указывать расширение имени файла `.htm` или `.html`. Для того, что бы можно было увидеть отредактированный HTML-документ, необходимо не забывать сохранять его в программе текстового редактора и обновлять содержимое окна браузера.

Однако, иногда при просмотре Web-страницы в окне браузера отображается не читабельный текст, а «кракозябры». Так происходит потому, что некоторые браузеры автоматически неправильно определяют кодировку текста HTML документа. Для корректного отображения Web-страницы в браузере необходимо выполнить команду **Вид ⇨ Кодировка** и указать кодировку, в которой она была изначально создана. Чтобы всякий раз при несовпадении кодировок в текстовом редакторе и браузере не выполнять эту команду необходимо один раз установить в нем предпочитаемую кодировку.

Аналогичная ситуация иногда наблюдается также при переносе HTML-документов из Windows в Linux или наоборот – из Linux в Windows. По умолчанию текст в операционной системе Linux кодируется по таблице символов **Юникод (UTF-8)**, а в Windows – **Кириллица (Windows-1251)**. Поэтому для обеспечения единообразия необходимо сохранять текст HTML документов и в Linux и в Windows в одной кодировке. Например, при сохранении HTML-документа в программе Блокнот явно указывать кодировку – **UTF-8**. Самым же простым и надежным способом добиться правильного отображения символов в браузере является использование тега `<meta>`. Он записывается в самом начале HTML-документа (в области заголовка `<head>`) и указывает, в какой кодировке изначально он был создан. Так, для Linux он имеет вид – `<meta charset=utf-8>`, а для Windows – `<meta charset=windows-1251>`, соответственно.

Практически все современные браузеры по умолчанию настроены на использование своей внутренней программы просмотра кода страниц, которая позволяет только просматривать, но не редактировать HTML-код. Однако, в большинстве случаев всегда можно настроить браузер так, чтобы для просмотра и редактирования HTML-кода страницы он запускал необходимую внешнюю программу.

Любой документ HTML состоит из текста, который необходимо отобразить, и инструкций, которые управляют способом отображения. Такие инструкции называются *тегами*. Чтобы браузер мог отличить текст, который необходимо отобразить, от тегов последние заключаются в символы «<» «>». Текст HTML документа должен начинаться тегом `<html>`, а заканчиваться `</html>`. Внутри этих тегов помещается «голова» и «тело» документа, которые обозначаются тегами `<head>...</head>` и `<body>...</body>`, соответственно. Тег `<title>...</title>`, помещаемый внутри тега `<head>...</head>`, служит для указания информации, которая будет отображаться в заголовке окна браузера. Информация, которая должна отображаться в основном окне браузера, помещается внутри тега `<body>...</body>`. Большинство браузеров не чувствительны к регистру символов в тегах, поэтому их можно записывать как строчными, так и прописными буквами, или их сочетанием. Например, правильно будут обработаны такие теги, как `<BODY>`, `<body>` и `<Body>`.

Цвет фона и символов HTML-документа можно указать двумя способами:

1. в формате **RGB**, когда он указывается долями трех его составляющих:

- **красной (R)** – от англ. **Red**,



- **зеленой (G)** – от англ. **Green** и
- **синей (B)** – от англ. **Blue** или

2. его английским названием.

При указании цвета в формате **RGB** он получается в результате сложения трех его компонентов. Интенсивность каждой из этих составляющих указывается своим 16-ричным значением в диапазоне от 00 до FF – чем больше число, тем интенсивнее этот цвет в результирующем. Примеры записи цветов приведены в таблице 1.

Таблица 1

Название цвета			Название цвета		
Русское	Английское	RGB	Русское	Английское	RGB
черный	black	000000	фиолетовый	purple	FF00FF
белый	white	FFFFFF	желтый	yellow	FFFF00
красный	red	FF0000	коричневый	brown	996633
зеленый	green	00FF00	оранжевый	orange	FF8000
бирюзовый	azure	00FFFF	лиловый	violet	8000FF
синий	blue	0000FF	серый	gray	A0A0A0

Цвет фона устанавливается при помощи атрибута `bgcolor=` в теге `<body>`. Если он не указан, то по умолчанию используется цвет, указанный в настройках браузера. Как правило – это белый цвет.

Цвет текста, отображаемого в окне браузера по умолчанию, – черный. Он указывается в его настройках. Явно цвет текста можно указать в теге `<body>` при помощи атрибута `text=`. Цвет отдельного фрагмента текста изменяется при помощи атрибута `color=` тега `<font>`.

Все, с теорией закончили. Настала пора практики. В конце концов, учиться лучше всего на примерах. И дальше – по инструкции:

## ***Ход работы***

### **Шаг 1 Настройка инструментария и создание шаблона Web-страницы**

Чтобы создать HTML-документ, выполните следующее:

1. В своем рабочем каталоге, например:

- `/home/student1/ivanov/` (для Linux) или

- D:\course1\БУА-1\Иванов\ (для Windows) или
- указанном преподавателем

создайте каталог html, в котором будут храниться документы HTML, создаваемые по ходу выполнения лабораторных работ.

2. В созданном ранее каталоге html создайте каталог lr1 и сделайте его текущим.

В каталоге lr1 будут храниться документы HTML, создаваемые по ходу выполнения текущей лабораторной работы.

3. В каталоге lr1 на свободном месте щелкните правой кнопкой мыши и в раскрывшемся контекстном меню выполните команду:

- **Создать документ ⇨ Пустой файл** (для Linux) или
- **Создать ⇨ Текстовый документ (Plain Text)** (для Windows)

и в раскрывшемся окне диалога укажите в качестве имени файла template.html.

В случае выдачи запроса на изменение расширения имени файла – подтвердите.

4. Щелкните правой кнопкой мыши по вновь созданному файлу template.html и в раскрывшемся контекстном меню выполните команду:

- **Открыть с помощью "Текстовый редактор Mousepad"** или
- **Открыть с помощью ⇨ Открыть с помощью "Текстовый редактор Mousepad"** (для Linux) или
- **Открыть с помощью а Блокнот (Notepad)** (для Windows).

Если указанные редакторы отсутствуют в контекстном меню, то их необходимо найти самостоятельно при помощи последнего пункта:

- **Открыть с помощью другого приложения...** (для Linux) или
- **Выбрать программу...** (для Windows).

5. В раскрывшемся окне редактора наберите следующий код (текст):

```
<html>
  <head>
    <meta charset=utf-8>
    <title></title>
  </head>
  <body>
  </body>
</html>
```

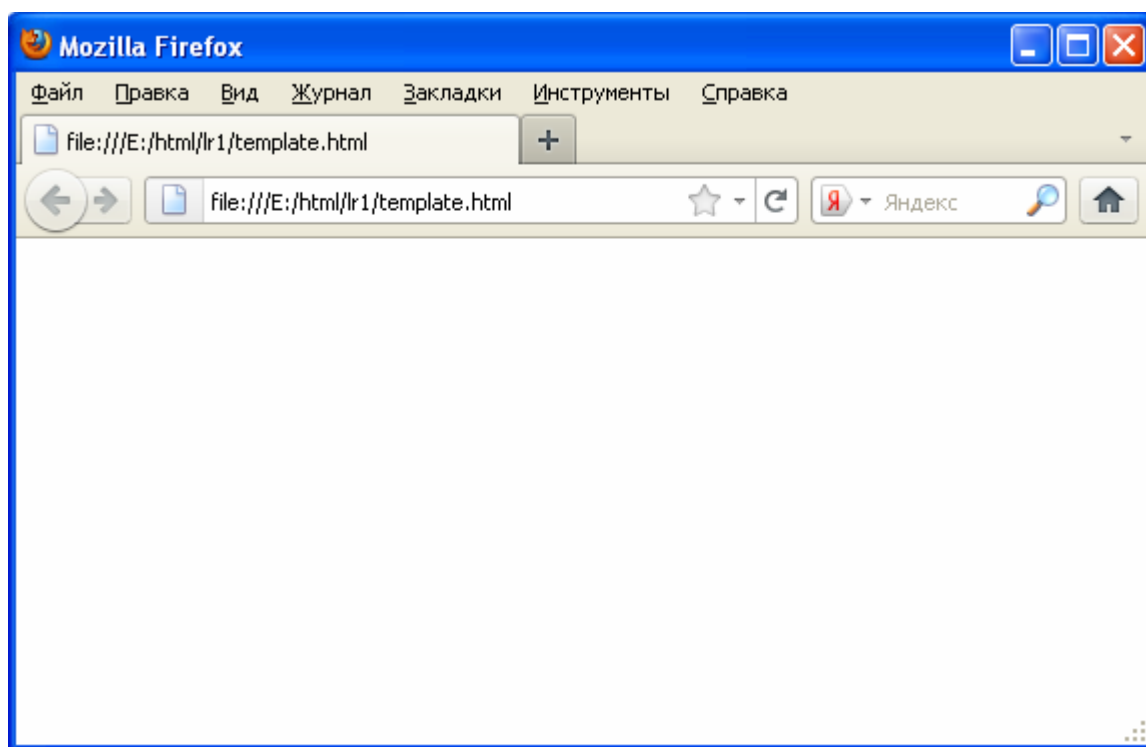
6. Сохраните файл, выполнив команду **Файл ⇒ Сохранить**.
7. Для просмотра созданной Web-страницы щелкните правой кнопкой мыши по файлу `template.html` и в раскрывшемся контекстном меню выполните команду:

- **Открыть с помощью "Интернет-проводник Firefox"** или
- **Открыть с помощью ⇒ Открыть с помощью "Интернет-проводник Firefox"** (для Linux) или
- **Открыть с помощью a Internet Explorer** (для Windows) или
- выберите любой другой браузер, установленный в Вашей системе.

Если необходимый браузер отсутствует в меню, то его необходимо найти самостоятельно при помощи последнего пункта:

- **Открыть с помощью другого приложения...** (для Linux) или
- **Выбрать программу...** (для Windows).

В результате откроется пустое окно браузера:



Так происходит потому, что созданный файл не содержит никакой полезной информации, а является всего лишь заготовкой (шаблоном) для любого HTML-документа, т. е. в дальнейшем его можно будет брать за основу при создании произвольной Web-страницы.

8. В текстовом редакторе отредактируйте файл `template.html` так, чтобы между тегами `<title>` и `</title>` был помещен текст «Мой первый документ HTML», а между `<body>` и `</body>` – «Всем огромный привет!».

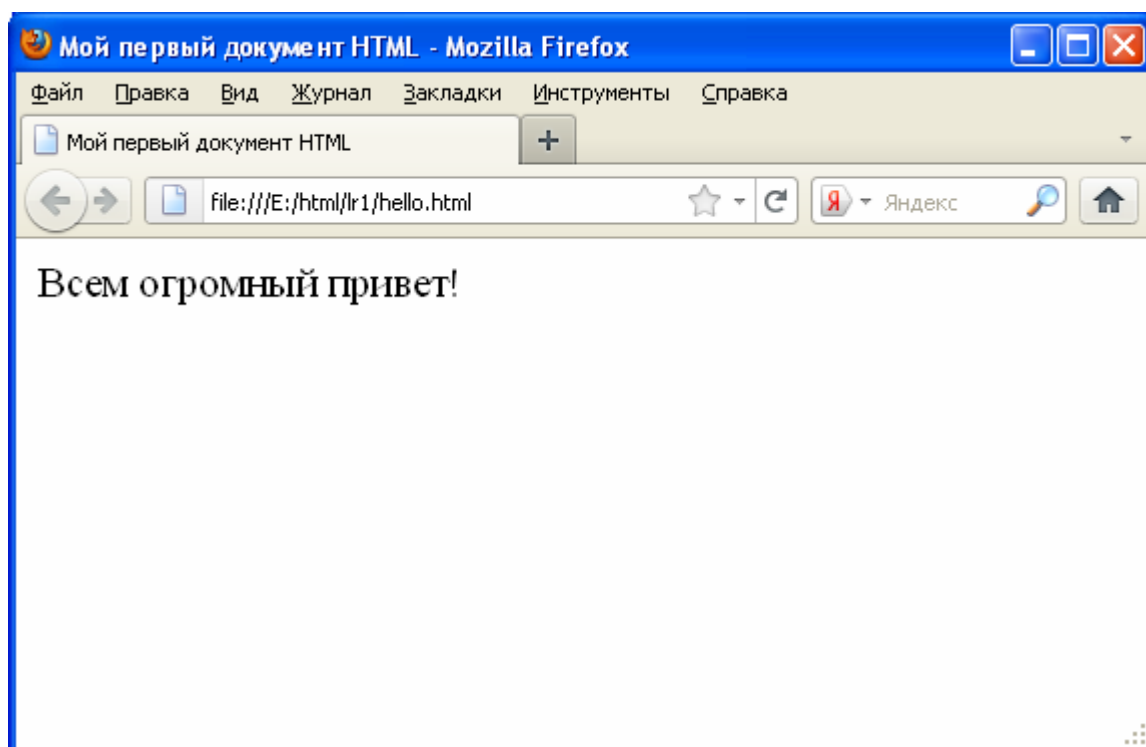
В результате документ HTML должен иметь такое содержание:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    Всем огромный привет!
  </body>
</html>
```

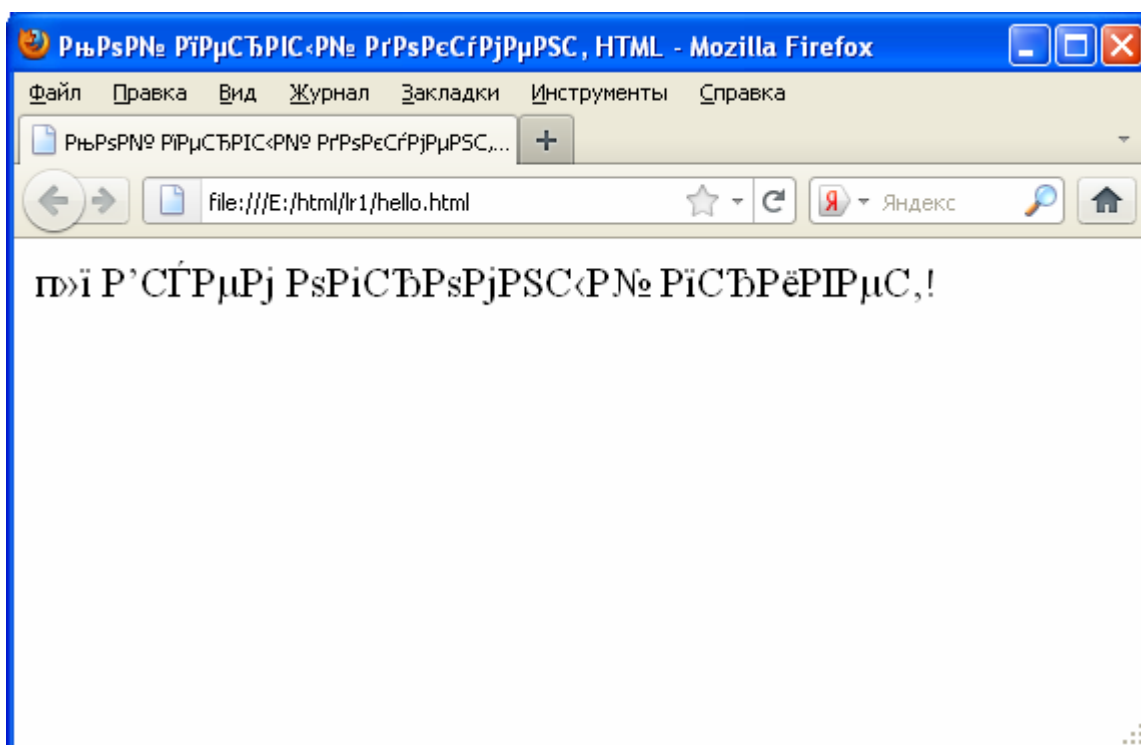
9. Сохраните отредактированный документ в файле `hello.html`, выполнив команду **Файл ⇨ Сохранить как....**
10. Закройте браузер.
11. Щелкните правой кнопкой мыши по вновь созданному файлу `hello.html` и в раскрывшемся контекстном меню выберите команду **Свойства**.
12. В раскрывшемся окне диалога свойств файла `hello.html`:
- на вкладке **Основные** при помощи раскрывающегося списка **Открывать с помощью:** укажите, например, **Интернет-проводник Firefox** (для Linux) или
  - на вкладке **Общие** в групповой рамке **Тип файла: Приложение:** щелкните по кнопке **Изменить** и в раскрывшемся далее окне диалога **Выбор программы** укажите, например, **Internet Explorer** (для Windows).

13. Выполните двойной щелчок по файлу `hello.html`.

В результате будет запущен, выбранный в предыдущем пункте, браузер и Вы увидите примерно следующее:



Если же в окне браузера вместо читабельного текста будут отображаться «кракозябры» как, например:



то это лишь означает, что браузер не смог автоматически определить кодировку русского (кириллического) текста. Для правильного отображения текста необходимо в браузере выполнить команду **Вид ⇨ Кодировка** и выбрать:

- **Юникод (UTF-8)** – если текст Web-страницы был набран в Linux или
- **Кириллица (Windows-1251)** – если текст Web-страницы был набран в Windows.

Такая «ошибка» может возникнуть также в ситуации когда, например, текст Web-страницы был подготовлен в операционной системе Windows, а просматривается браузером в Linux или, наоборот. По умолчанию текст в операционной системе Linux кодируется в кодировке **Юникод (UTF-8)**, а в Windows – **Кириллица (Windows-1251)**. Для обеспечения единообразия в такой ситуации рекомендуется при сохранении файла программой Блокнот (Notepad) в операционной системе Windows в окне диалога **Сохранить как** в раскрывающемся списке **Кодировка** выбрать значение **UTF-8**.

При редактировании текста, набранного в Windows, без перекодировки в текстовом редакторе Mousepad операционной системы Linux необходимо при его открытии правильно указать исходную кодировку. Для этого в переключателе кодировок установить значение **Иная**, а в раскрывающемся списке – **WINDOWS-1251**.

Чтобы всякий раз при несовпадении кодировок в текстовом редакторе и браузере не выполнять команду **Вид ⇨ Кодировка** необходимо установить в нем предпочитаемую кодировку. Для этого, например, в браузере Firefox в окне диалога **Настройки** на вкладке **Содержимое** в групповой рамке **Шрифты и цвета** щелкните по кнопке **Дополнительно...** и в раскрывшемся окне диалога **Шрифты** при помощи раскрывающегося списка **Кодировка по умолчанию** выставьте необходимое значение, например, **Юникод (UTF-8)**. Также правильному распознаванию кодировки русского текста способствует правильная установка значения **Русская** (или, наоборот, **(Отключено)**) по команде **Вид ⇨ Кодировка ⇨ Автоопределение**.

Самым же простым и надежным способом добиться правильного отображения символов в браузере является использование тега `<meta>`. Он записывается в самом начале HTML-документа (в области заголовка `<head>`) и указывает, в какой кодировке изначально он был создан. Так, для Linux он имеет вид – `<meta charset=utf-8>`, а для Windows – `<meta charset=windows-1251>`, соответственно.

## Шаг 2 Настройка редактора и задание цвета текста

Чтобы установить цвет текста ..., выполните следующее:

1. Закройте программу текстового редактора.
2. В окне браузера на свободном месте щелкните правой кнопкой мыши и в раскрывшемся контекстном меню выберите пункт **Исходный код страницы** (или **Просмотр HTML кода** или **Просмотр кода страницы**) или в основном меню **Вид** (или **Инструменты** ⇨ **Веб-разработка**) выберите аналогичный пункт.

В результате в большинстве современных браузеров откроется дополнительное окно, в котором будет исходный HTML-код страницы, редактировать который нельзя – его можно только просматривать.

Поэтому, ... есть два варианта:

- настроить браузер так, чтобы для просмотра и редактирования кода страницы он запускал необходимый внешний редактор (далее, пункт 4), или
  - для редактирования Web-страницы запускать внешний редактор через файловый менеджер (Проводник, Thunar и т. д.) по щелчку на нем правой кнопки мыши (далее, пункт 5).
3. Закройте окно с HTML-кодом.
  4. Настройте браузер так, чтобы для просмотра и редактирования кода Web-страницы он запускал необходимый внешний редактор.

Для этого:

- в *Internet Explorer*: выполните команду **Сервис** ⇨ **Средства разработчика** (или **F12**), и в раскрывшемся окне выполните команду **Файл** ⇨ **Настроить источник представлений Internet Explorer**. А далее, выберите **Блокнот** – чтобы править код в Блокноте, или **Прочие ...** – чтобы подключить любой другой внешний редактор.
- в *Firefox*: перейдите по адресу `about:config`; при этом откроется окно диалога с предупреждением об осторожном изменении настроек. На открывшейся странице в поле редактирования **Фильтр (Поиск)** введите `view_source.editor`. В результате отображаемый список параметров уменьшится только для тех, которые соответствуют шаблону (фильтру). Затем необходимо установить значение параметра `view_source.editor.external` в `true`, а в `view_source.editor.path` – записать путь к требуемому редактору. Например, `C:\WINDOWS\system32\notepad.exe` для программы

Блокнот в Windows, или – /usr/bin/mousepad – для Mousepad в Linux.

Теперь для редактирования Web-страниц по щелчку правой кнопкой мыши в окне браузера можно будет запускать выбранный редактор.

Или – второй вариант:

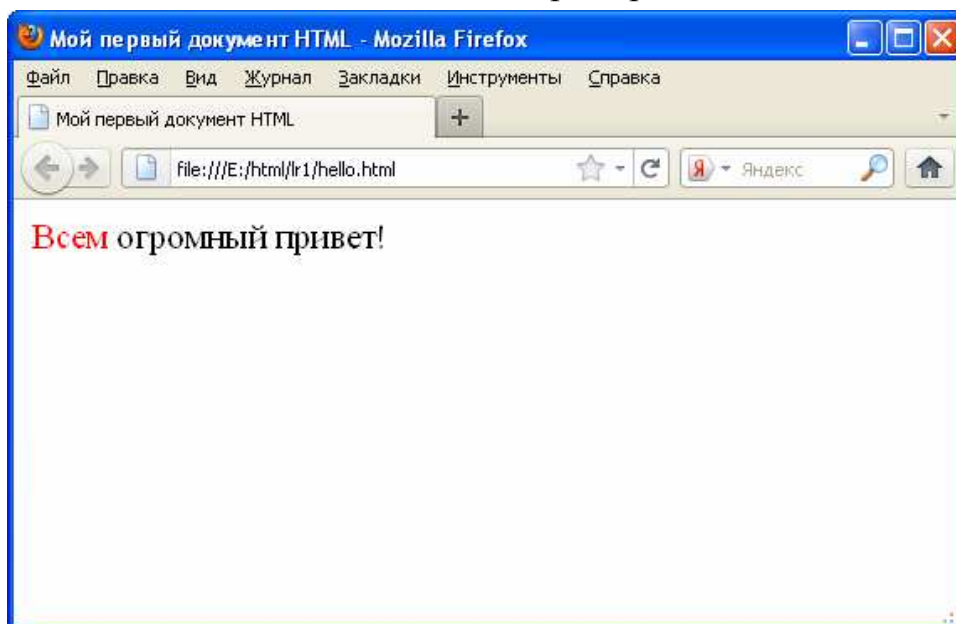
5. Выполните щелчок правой кнопкой мыши по файлу hello.html, а в раскрывшемся контекстном меню выберите пункт **Открыть с помощью** и укажите необходимый текстовый редактор.
6. Измените цвет слова «Всем» на **красный**, указав название цвета.

Для этого файл hello.html отредактируйте следующим образом:

```
<html>
<head>
  <meta charset=utf-8>
  <title>Мой первый документ HTML</title>
</head>
<body>
  <font color="red">Всем</font> огромный привет!
</body>
</html>
```

7. Сохраните внесенные в файл hello.html изменения, выполнив команду **Файл ⇨ Сохранить**.
8. Обновите содержимое окна браузера нажав, соответственно, кнопку **Обновить (Обновить текущую страницу, Обновить эту страницу)** или клавишу **F5**.

В результате его окно должно выглядеть примерно так:





Если Web-страница сильно отличается от представленной, то повторите три последних пункта еще раз.

9. Отредактируйте файл `hello.html` так, чтобы цвет слова «огромный» был **зеленым**, указав цвет в формате **RGB**:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <font color="red">Всем</font> <font
color="#00FF00">огромный</font> привет!
  </body>
</html>
```

Обратите внимание, что значению цвета обязательно должен предшествовать знак «решетка» – «#», а в самом коде – указано максимальное значение для зеленой компоненты (FF), а две другие – красная и синяя – имеют минимальные значения (00).

10. Просмотрите отредактированную Web-страницу.

Для этого:

- а.** сохраните ее в текстовом редакторе и
- б.** обновите содержание окна браузера.

11. Установите весь остальной текст на странице **синим**.

Для этого отредактируйте файл `hello.html` следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body text="#0000FF">
    <font color="red">Всем</font> <font
color="#00FF00">огромный</font> привет!
  </body>
</html>
```

12. Просмотрите в браузере отредактированную Web-страницу.

### Шаг 3 Задание цвета фона

Чтобы установить цвет фона выполните следующее:

1. Отредактируйте файл `hello.html` следующим образом и просмотрите его в браузере:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body text="#0000FF" bgcolor="#000000">
    <font color="red">Всем</font> <font
color="#00FF00">огромный</font> привет!
  </body>
</html>
```

### Индивидуальные задания

1. В созданном ранее каталоге `lr1` создайте каталог `iz`.  
В нем будут храниться документы HTML, создаваемые по ходу выполнения индивидуального задания текущей лабораторной работы.
2. В соответствии с номером по списку группы, или указанием преподавателя, установите такие цвета на Web-странице, а так же обратите внимание на их сочетаемость и уместность:

Вариант	Фон	Текст	Слово «Всем»	Слово «огромный»
1	2	3	4	5
1	белый	черный	серый	синий
2	черный	белый	красный	серый
3	черный	зеленый	серый	красный
4	синий	белый	зеленый	бирюзовый
5	лиловый	белый	желтый	оранжевый

Продолжение

1	2	3	4	5
6	серый	зеленый	лиловый	бирюзовый
7	красный	фиолетовый	лиловый	зеленый
8	фиолетовый	зеленый	серый	синий
9	зеленый	синий	красный	серый
10	серый	черный	зеленый	лиловый
11	лиловый	зеленый	фиолетовый	черный
12	коричневый	зеленый	серый	красный
13	синий	серый	зеленый	черный
14	черный	зеленый	серый	лиловый
15	оранжевый	лиловый	фиолетовый	белый
16	зеленый	синий	оранжевый	серый
17	лиловый	зеленый	черный	синий
18	бирюзовый	черный	белый	коричневый
19	оранжевый	фиолетовый	лиловый	серый
20	серый	зеленый	оранжевый	коричневый
21	коричневый	белый	зеленый	красный
22	красный	зеленый	черный	коричневый
23	белый	оранжевый	зеленый	фиолетовый
24	фиолетовый	бирюзовый	лиловый	черный
25	черный	фиолетовый	оранжевый	серый
26	серый	оранжевый	фиолетовый	красный
27	оранжевый	фиолетовый	красный	серый
28	желтый	лиловый	фиолетовый	бирюзовый
29	бирюзовый	красный	коричневый	лиловый
30	зеленый	синий	оранжевый	черный

3. Созданную Web-страницу сохраните в каталоге i z под именем my.html.

4. В тетрадь для выполнения лабораторных работ занесите эскиз созданной Web-страницы и, соответствующей ей, текст (код) HTML-документа.

### ***Контрольные вопросы***

1. С помощью какой программы, в простейшем случае, можно подготовить HTML-документ в Windows и Linux, соответственно?
2. Как в программе текстового редактора сохранить созданный HTML-документ?
3. С помощью какой программы можно просмотреть созданную Web-страницу?
4. Почему в браузере могут не отображаться изменения HTML-документа, выполненные в программе текстового редактора?
5. Почему иногда браузер вместо читабельного текста отображает «кракозябры»?
6. Как настроить браузер, чтобы он правильно отображал текст Web-страницы?
7. В какой кодировке по умолчанию кодируется текст в операционной системе Windows, а в какой – в Linux?
8. Как необходимо переносить файлы HTML-документов из Windows в Linux, и наоборот, чтобы они правильно отображались в браузере?
9. Как необходимо настроить файловый менеджер, чтобы при двойном щелчке по файлу HTML-документа он открывался в нужной программе-браузере?
10. Как необходимо настроить браузер, чтобы при просмотре исходного HTML-кода страницы его можно было редактировать?
11. Для чего предназначен язык HTML?
12. Для чего служат теги?
13. Как браузер отличает теги от остального текста документа?
14. Из каких обязательных частей состоит документ HTML?
15. Какие теги обязательно присутствуют в любом документе HTML?
16. Как установить заголовок окна браузера?
17. Где записывается информация, которая должна отображаться в основном окне браузера?
18. Для чего служит тег `<html>`?
19. Для чего служит тег `<head>`?
20. Для чего служит тег `<title>`?
21. Для чего служит тег `<body>`?
22. Какие атрибуты может иметь тег `<body>`?

23. Как браузеры отображают текст и теги в различных регистрах?
24. Сколько есть способов указания цвета фона и текста HTML-страницы?
25. Как указать цвет в формате RGB?
26. Как установить цвет отдельного фрагмента текста на Web-странице?
27. Как установить цвет всего текста на Web-странице?
28. Какой цвет текста на Web-странице используется по умолчанию?
29. Как установить цвет фона страницы HTML?
30. Какой цвет фона на Web-странице используется по умолчанию?
31. Сформулируйте свой вопрос по пройденному материалу.
32. Сформулируйте свое предложение по совершенствованию темы.

## Лабораторная работа № 2

### Форматирование Web-страниц

#### *Цель работы*

Освоение и приобретение практических навыков форматирования Web-страниц, строк и символов.

#### *Краткие теоретические сведения*

При обработке HTML-документов браузеры автоматически размещают текст на экране, не принимая во внимание встречающиеся в файле переводы строк и идущие подряд пробелы. Для «разлома» строк служит тег перевода строки `<br>` – он выполняет переход на новую строку. Тэг абзаца `<p>` тоже отделяет строку, но еще добавляет пустую строку, которая зрительно выделяет абзаца.

Тэги выделения фрагментов текста позволяют управлять отображением отдельных символов и слов. Существует три тэга выделения фрагментов текста:

- `<b>...</b>` – для выделения **полужирным**;
- `<i>...</i>` – для выделения *курсивом*;
- `<u>...</u>` – для выделения подчеркиванием.

Большинство тегов, используемых для разметки HTML-страниц, являются контейнерами, т. е. внутри себя они могут содержать текст и другие теги. Например, использование комбинированных начертаний шрифтов:

`<i><b>Всем</b></i>` `<i>огромный</i>` `<u>привет!</u>`.

При этом необходимо помнить следующее правило записи вложенных тэгов: тег, который был открыт первым, закрывается последним, вторым – предпоследним и т. д. Говоря иначе, теги со всем их содержимым должны полностью вкладываться в другие теги, не оставляя «хвостов» снаружи. Например,

`<Тэг1> <Тэг2> <Тэг3> ... </Тэг3> </Тэг2> </Тэг1>` правильная запись

`<Тэг1> <Тэг2> <Тэг3> ... </Тэг3> <Тэг1> </Тэг2>` ошибочная запись

Тег, в который непосредственно вложен данный тег, называется *родительским*, или *родителем*. В свою очередь, тег, вложенный в данный тег, называется *дочерним*, или *потомком*. Уровень вложенности тегов в HTML-коде

обозначают с помощью отступов, которые ставят слева от соответствующего тега и создают с помощью пробелов. На отображение Web-страницы они никак не влияют.

Теги вместе с объектами (фрагментами документа), отображением которых он управляет, называются *элементами*. Формальный синтаксис элемента HTML-документа представлен на рисунке 1.

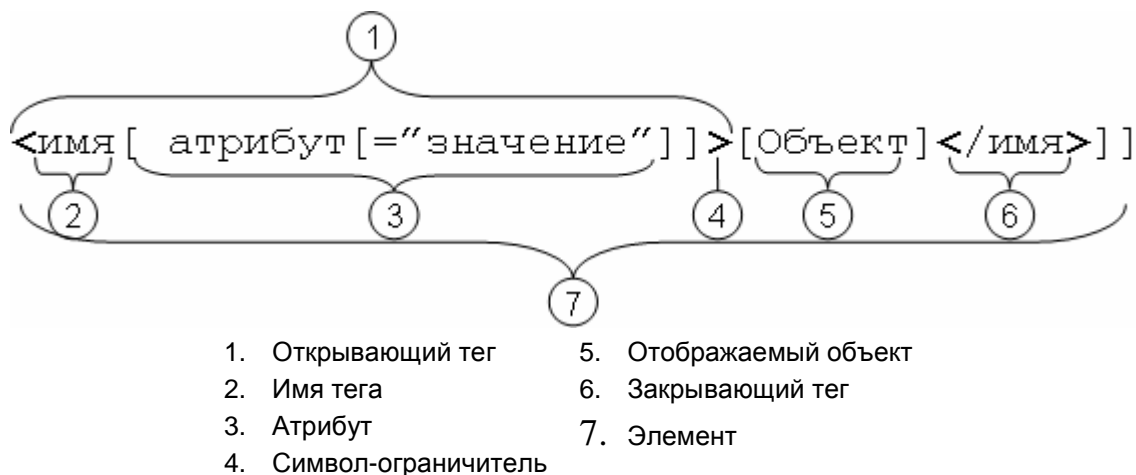


Рисунок 1 – Элемент HTML-документа

**!** Составляющие в квадратных скобках «[]» являются не обязательными.

Имя тега указывает на характер выполняемых им действий по разметке и форматированию, а атрибуты (параметры) уточняют способ выполнения этих действий. Атрибуты являются не обязательными составляющими тега и если они имеются, то отделяются от имени, и друг от друга, по крайней мере, одним пробелом. Порядок следования атрибутов тега произволен.

Существует два способа управления размером текста, отображаемого браузером:

- использование стилей заголовка,
- задание размера текущего шрифта.

В языке HTML имеется шесть тэгов заголовков (от <h1> до <h6>). Каждому такому тэгу соответствует свой стиль, заданный в параметрах настройки браузера. Из них <h1> – самый крупный.

Атрибут `size=` тега <font> позволяет задавать размер текущего шрифта для отдельных фрагментов текста. Диапазон установки размера текущего шрифта – от 1 до 7.

Тэг <font> многофункциональный – он предоставляет возможности управления размером, начертанием и цветом текста. Изменение гарнитуры

шрифта выполняется простым добавлением к тэгу <font> атрибута face=. Например, для отображения текста шрифтом Arial необходимо указать <font face="arial">. Для изменения цвета шрифта в тэге <font> используется атрибут color=.

С помощью тэгов языка HTML можно управлять горизонтальным выравниванием текста. Если не оговаривать способ выравнивания, то все элементы в документе будут выравниваться по левому краю и иметь неровное правое поле. Для выравнивания текста используется атрибут align=, который встраивается в теги абзаца <p> или заголовка <h> и может принимать такие значения:

1. center – выравнивание по центру;
2. right – выравнивание по правому краю;
3. left – выравнивание по левому краю.

Иногда на Web-странице необходимо отобразить символы, которых нет на клавиатуре, например, «знак зарегистрированной торговой марки» – (®) или символы, которые имеют специальное значение, например, «<» и «>» – начало и окончание тега. В этих случаях можно вводить нужные символы с помощью специальных кодов. Такие коды начинаются с символа «амперсанд» (&). За ним следует название символа либо его числовой код в десятичной или шестнадцатеричной системе. Завершает код символ «точка с запятой» (;). Десятичный код начинается символом «решетка» (#), а шестнадцатеричный – двумя – «решетка» и «латинская х» (#x).

Так, например, символ «<» определяется при помощи кода «&lt;» («&#60;» или «&#x3C;»), а символ «>» – кода «&gt;» («&#62;» или «&#x3E;»). Символ «знак зарегистрированной торговой марки» (®) имеет код «&reg;» («&#174;» или «&#xAE;»), «знак copyright» (©) – код «&copy;» («&#169;» или «&#xA9;»), а «неразрывный пробел» – «&nbsp;» («&#160;» или «&#xA0;»). Полный же список таких кодов имеется в любом справочнике по языку HTML, например, в «HTML-справочник» Владимира Городулина – <http://html.manual.ru/> или в «HTML в примерах» Александра Климова – <http://webmaster.alexanderklimov.ru>.

Все, с теорией закончили. Настала пора практики. В конце концов, учиться лучше всего на примерах. И дальше – по инструкции.



## ***Ход работы***

### **Шаг 1 Управление расположением текста на экране**

1. В каталоге для выполнения лабораторных работ по HTML – html – создайте каталог lr2 и сделайте его текущим.  
В нем будут храниться документы HTML, создаваемые по ходу выполнения текущей лабораторной работы.
2. Из каталога Лабораторной работы №1 – lr1 – скопируйте файл hello.html. Если его там нет – создайте заново.  
В результате в каталоге lr2 должен быть файл hello.html – результат выполнения предыдущей Лабораторной работы.
3. Отредактируйте файл hello.html, убрав в нем информацию о цвете и расположив слова «Всем», «огромный» и «привет!» на разных строчках:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    Всем
    огромный
    привет!
  </body>
</html>
```

4. Просмотрите его содержимое в браузере.  
В результате текст в основном окне будет расположен в одну строку. Так происходит потому, что при отображении HTML-документов браузеры автоматически размещают текст на экране, не принимая во внимание встречающиеся в файле переводы строк и идущие подряд пробелы.

## Шаг 2 Тэги перевода строки и абзаца

1. Внесите в текст файла HTML следующие изменения:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    Всем<p>
    огромный<br>
    привет!
  </body>
</html>
```

2. Посмотрите обновленную WEB-страницу.

В результате она должна выглядеть примерно так:



Вид документа HTML изменился в результате того, что было добавлено два тега: `<br>` и `<p>`. Тэг перевода строки `<br>` отделяет строку от последующего текста или графики, а `<p>` – дополнительно добавляет пустую строку.

### Шаг 3 Выделение фрагментов текста

1. Отредактируйте текст файла HTML следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <i><b>Всем</b></i> <i>огромный</i> <u>привет!</u>
  </body>
</html>
```

2. Посмотрите обновленную Web-страницу.

### Шаг 4 Использование стилей заголовка

1. Внесите в файл hello.html такие изменения:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <h1>Всем</h1> <i>огромный</i> <u>привет!</u>
  </body>
</html>
```

2. Посмотрите полученную Web-страницу.

## Шаг 5 Задание размера текущего шрифта

1. Отредактируйте файл `hello.html` следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <font size=7>Всем</font>
    огромный привет!
  </body>
</html>
```

2. Самостоятельно измените размер шрифта для текста «огромный привет!», используя тэг `<font>`, а так же, тэги выделения фрагментов текста и тэги перевода строки и абзаца что бы получить такую Web-страницу:

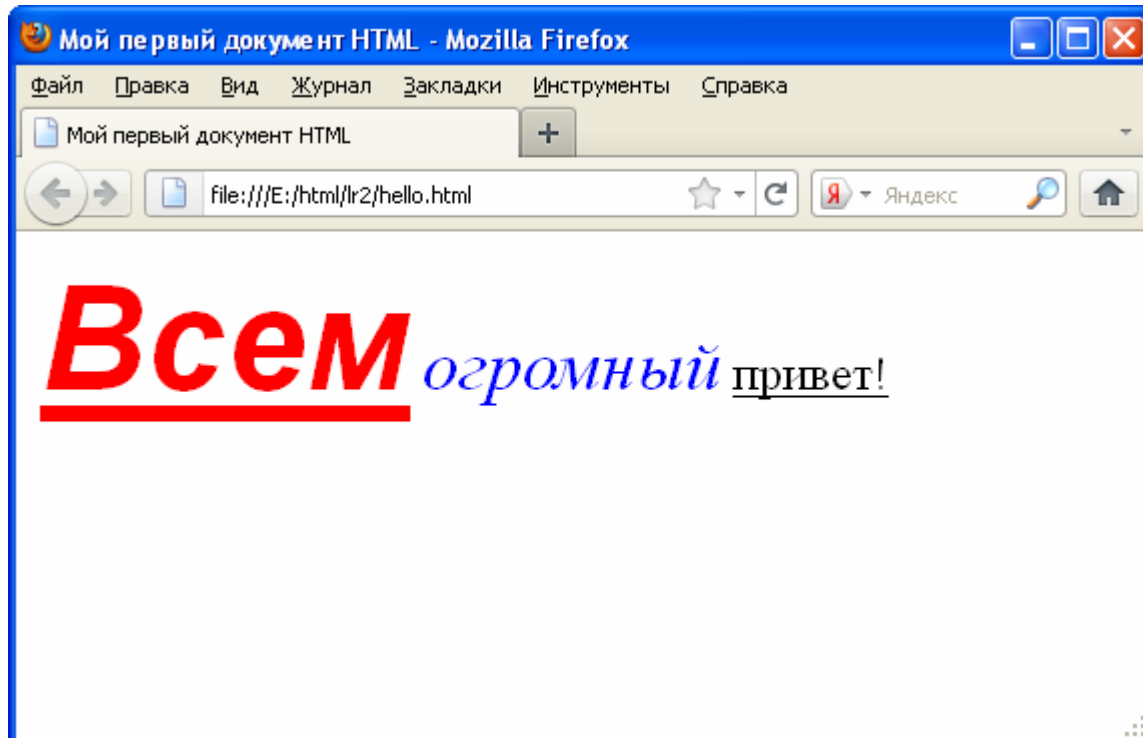


## Шаг 6 Гарнитура и цвет шрифта

1. Внесите в файл `hello.html` такие изменения:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <u><i><b><font color="#ff0000" face="arial" size="7">
      Всем</font></b></i></u> огромный привет!
    </body>
</html>
```

2. Самостоятельно измените размер, цвет, гарнитуру, стиль текста документа, так что бы он выглядел следующим образом:



## Шаг 7 Выравнивание текста по горизонтали

1. Отредактируйте файл `hello.html` следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <p align="center">
      <font color="#008080" size="7"><b>Всем</b></font><br>
      <font size="6"><i>огромный привет!</i></font>
    </p>
  </body>
</html>
```

2. Посмотрите полученный результат.
3. Проведите экспериментальную работу с созданным документом.  
Меняя размеры окна, посмотрите, как браузер показывает текст с принудительным разрывом строк. Что происходит, когда окно уменьшается настолько, что в нем не может поместиться целиком даже одно слово?

## Шаг 8 Отображение специальных символов

1. Отредактируйте файл `hello.html` так, чтобы слова «огромный привет!» всегда отображались в одной строке.  
Для этого замените пробел между словами «огромный» и «привет!» на специальный символ «неразрывный пробел» – «&nbsp;». В результате файл `hello.html` должен иметь такое содержание:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <p align="center">
      <font color="#008080" size="7"><b>Всем</b></font><br>
      <font size="6"><i>огромный&nbsp;привет!</i></font>
    </p>
  </body>
</html>
```

2. Посмотрите обновленную Web-страницу.
3. В конце Web-страницы `hello.html` перечислите все теги, которые были использованы при ее форматировании.

Для этого файл `hello.html` отредактируйте, например, следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <p align="center">
      <font color="#008080" size="7"><b>Всем</b></font><br>
      <font size="6"><i>огромный&nbsp;привет!</i></font>
    </p>
    Для форматирования этого текста были использованы
    следующие теги:<br>
    &lt;p&gt; ,<br>
    &#60;br&#62; ,<br>
    &#x3C;font&#x3E; ,<br>
    &lt;b&gt; и<br>
    &lt;i&gt; .<br>
  </body>
</html>
```

4. Посмотрите полученную Web-страницу.

### ***Индивидуальные задания***

1. В созданном ранее каталоге `lr2` создайте каталог `iz`.  
В нем будут храниться документы HTML, создаваемые по ходу выполнения индивидуального задания текущей лабораторной работы.
2. В каталоге `iz` создайте свой документ HTML и сохраните его в файле под именем `index.html`.

Пусть это будет небольшое приветствие с рассказом, например, о себе, родителях, увлечениях и т. д. В заголовок окна поместите Вашу Фамилию Имя и Отчество.

3. В тетрадь для выполнения лабораторных работ занесите эскиз созданной Web-страницы и, соответствующей ей, текст HTML-документа.

### **Контрольные вопросы**

1. Как браузеры обрабатывают символы пробелов и новых строк?
2. Как выполнить переход на новую строку на странице HTML?
3. Объясните назначение тега `<p>`.
4. Объясните назначение тега `<br>`.
5. Как установить полужирное начертание символов?
6. Как установить подчеркивание символов?
7. Как установить начертание символов курсив?
8. Как можно одновременно изменить несколько параметров начертания символов?
9. Как записываются вложенные теги?
10. Объясните структуру элемента документа HTML.
11. Объясните структуру тега.
12. Как отделяются друг от друга составляющие тега?
13. Для чего служат атрибуты тега?
14. Каков порядок следования атрибутов тега?
15. Сколько есть способов изменения размеров символов?
16. Для чего служат теги `<h1>` – `<h6>`?
17. Какой заголовок больше `<h1>` или `<h6>`?
18. Для чего служит тег `<font>`?
19. Какие атрибуты имеет тег `<font>`?
20. Как установить размер шрифта?
21. Как установить гарнитуру шрифта?
22. Как установить цвет выделенного фрагмента текста?
23. Как можно управлять горизонтальным выравниванием текста?
24. Какие значения может принимать атрибут `align`?
25. С какими тегами может использоваться атрибут `align`?
26. Что происходит с текстом при изменении размеров окна?
27. В чем разница между тегом `<br>` и тегом `<p>`?



28. Как отобразить на Web-странице символ, которого нет на клавиатуре или который имеет специальное значение (например, «<»)?
29. Объясните структуру кода для отображения специальных символов.
30. Перечислите способы построения кодов для отображения специальных символов.
31. Сформулируйте свой вопрос по пройденному материалу.
32. Сформулируйте свое предложение по совершенствованию темы.

## Лабораторная работа № 3

### Гиперссылки и графика

#### *Цель работы*

Освоение и приобретение практических навыков создания и работы с гиперссылками, а также вставки графических изображений в HTML-документ и использования их в качестве гиперссылок.

#### *Краткие теоретические сведения*

При хранении вся информация может быть представлена либо в виде одного цельного блока (как один большой текстовый документ) или в виде нескольких независимых блоков, связанных между собой только специально оформленными ссылками. Хранение и представление информации в виде независимых частей (страниц), связанных между собой ссылками, называется *гипертекстом* или *гипертекстовым документом*. Ссылки между частями называются *гипертекстовыми ссылками* или просто *ссылками*. Ссылки могут указывать, как на отдельные части документа, так и на различные документы. Ссылки в пределах одного документа называются *внутренними*, а на другие документы – *внешними*. Внешние ссылки могут быть *относительными* – если указывают на документ в пределах одного сайта (или компьютера) и *абсолютными* – когда указывается протокол обмена и полный адрес документа в сети. Если будет указана только каталог, без имени конкретного файла, то предполагается, что это ссылка на сайт, и в этом каталоге будет открыт файл `index.html`, `index.htm`, `default.html`, `default.htm` или `index.php` (конкретное имя указывается в настройках Web-сервера).

Создать текстовую гиперссылку очень просто. Достаточно найти в HTML-документе фрагмент текста, который нужно превратить в гиперссылку, и заключить его в парный тег `<a>`. Интернет-адрес целевой Web-страницы указывают в атрибуте `href=` этого тега.

При помощи атрибута `name=` тега `<a>` указывается место в документе, на которое затем можно будет перейти. Другой способ указания точки перехода в документе – при помощи общего атрибута `id=`. Имя точки перехода, указанной при помощи атрибута `name=` или `id=`, в атрибуте `href=` должно обязательно начинаться символом «#». Например, если `name="label1"`, то

href="#label1". Атрибут href= тега <a> позволяет также создавать гиперссылки на конкретные метки во внешних файлах. В этом случае имя метки отделяется от имени файла символом «#». Например, href="index.html#label1" – ссылка на метку label1 во внешнем файле index.html.

Правила отображения гиперссылок Web-обозревателем:

- при помещении курсора мыши на гиперссылку браузер меняет его форму на «указующий перст»;
- текст любых гиперссылок подчеркивается;
- обычные гиперссылки выделяются синим цветом;
- гиперссылки, по которым посетитель уже «ходил» (посещенные гиперссылки), выводятся темно-красным цветом;
- гиперссылка, по которой посетитель в данный момент щелкает (активная гиперссылка), выводится ярко-красным цветом.

Таково поведение по умолчанию.

Однако, цвет гиперссылок в документе HTML можно явно указать в теге <body> с помощью атрибутов:

- link= – просто ссылка,
- alink= – выделенная ссылка и
- vlink= – уже посещаемая ссылка.

С помощью необязательного атрибута target= тега <a> можно указать место загрузки ссылки:

- target="\_parent" – в текущее окно браузера, или
- target="\_blank" – в новое.

Атрибут title= тега <a> служит для создания подсказки, которая всплывает при выделении гиперссылки. Такая подсказка полезна, например, в случае, когда в качестве гиперссылки используется изображение.

Гиперссылки на адреса электронной почты указываются немного иначе, чем на другие документы, например, <a href="mailto:pochta@mail.ru">Мой e-mail</a>. Здесь «mailto:» записывается обязательно, и точно так, как указано. Подобным образом создаются гиперссылки на другие службы Internet, например, news-конференции, только вместо «mailto:» пишется «news:», а вместо адреса электронной почты – имя группы новостей.

В WWW сейчас используются всего три формата графических файлов:

- **GIF** (Graphics Interchange Format, формат обмена графикой) предназначен для хранения векторных (штриховых) изображений. Среди достоинств этого формата следует отметить то, что он позволяет задать для изображения «прозрачный» цвет, в одном GIF-файле можно хранить сразу несколько изображений, фактически – настоящий фильм (анимированный GIF). Недостаток у формата GIF всего один – он совершенно не годится для хранения полутоновых изображений (фотографий, картин и т. п.). Это обусловлено потерями качества при сжатии и тем, что GIF-изображения могут включать всего 256 цветов. Расширение имени файла формата GIF – .GIF.
- **JPEG** (Joint Photographic Experts Group, объединенная группа экспертов по фотографии) был разработан специально для хранения растровых (полутоновых) изображений. JPEG, в отличие от GIF, не ограничивает количество цветов у изображения, а реализованное в нем сжатие лучше всего подходит для полутоновых изображений. Однако он плохо справляется с штриховой графикой, не поддерживает «прозрачный» цвет и анимацию. Расширение имени файла формата JPEG – .JPEG, .JPG или .JPE.
- **PNG** (Portable Network Graphics, перемещаемая сетевая графика) был разрабатывался как замена формату GIF, а также, в некоторой степени, JPEG. В настоящее время он последовательно отвоевывает «жизненное пространство» у этих форматов. К достоинствам формата PNG можно отнести возможность хранения как штриховых, так и полутоновых изображений и поддержку полупрозрачности. Недостаток всего один и не критичный – невозможность хранения анимации. Расширение имени файла формата PNG – .PNG.

Графические объекты в HTML-документ вставляются при помощи тега `<img>`. Атрибут `src=` является обязательным и указывает имя файла, содержащего графическое изображение. Выравнивание изображения выполняется при помощи атрибута `align=`. Так значение атрибута `align=left` выравнивает изображение по левому краю окна, `align=right` – по правому. Рамка вокруг изображения устанавливается при помощи атрибута `border=`. Ее толщина указывается в пикселях. Атрибуты `height=` и `width=` указывают размер изображения в пикселях по высоте и ширине, соответственно. Всегда желательно явно устанавливать эти атрибуты для ускорения загрузки страницы браузером. Атрибуты `hspace=` и `vspace=`

указывают размер свободного места в пикселях слева и справа, и сверху и снизу от изображения, соответственно.

Язык HTML позволяет использовать в качестве содержимого гиперссылки любой фрагмент любого элемента, в том числе и графическое изображение, т. е. создавать изображение-гиперссылку. Для создания изображения-гиперссылки достаточно поместить внутри тега `<a>` тег `<img>`.

Все, с теорией закончили. Настала пора практики. В конце концов, учиться лучше всего на примерах. И дальше – по инструкции:

## ***Ход работы***

### **Шаг 1 Создание гиперссылок в пределах одного каталога**

1. В каталоге для выполнения лабораторных работ по HTML – `html`, создайте каталог `lr3` и сделайте его текущим.
2. Из каталога Лабораторной работы №2 – `lr2`, скопируйте файл `hello.html`. Если его там нет – создайте заново.  
В результате в каталоге `lr3` должен быть файл `hello.html` – результат выполнения предыдущей Лабораторной работы.
3. Создайте Web-страничку такого содержания и назовите ее `index.html`.


```
<html>
  <head>
    <meta charset=utf-8>
    <title>Главная страница</title>
  </head>
  <body>
    Приветствую Вас на своей страничке!<br><br>
    Вот мое <a href="hello.html">приветствие</a>
  </body>
</html>
```


4. Щелкните по ссылке на слове «приветствие».  
В результате должен открыться документ `hello.html` – копия из предыдущей Лабораторной работы.

## Шаг 2 Создание гиперссылок в пределах разных каталогов


1. В текущем каталоге создайте каталог `hello` и переместите туда файл `hello.html`.
2. В файле `index.html` замените строку:  
«Вот мое `<a href = "hello.html"> приветствие<a>`»  
на строку:  
«Вот мое `<a href = "hello/hello.html"> приветствие<a>`».  
Запись «`hello/hello.html`» говорит о том, что файл `hello.html` расположен во вложенном каталоге («этажом ниже») по отношению к текущему файлу – `index.html`.
3. Щелкните по ссылке на слове «приветствие».  
В результате должен снова открыться файл `hello.html`.
4. В конце файла `hello.html` добавьте такую строку:  
«`<a href = "../index.html">Вернуться к главной`  
странице`</a>`».  
Запись «`../index.html`» говорит о том, что файл `index.html` расположен в родительском каталоге («этажом выше») по отношению к текущему файлу – `hello.html`.
5. Проверьте работоспособность ссылок.

## Шаг 3 Вставка рисунков

1. В текущем каталоге `lr3` создайте рисунок `50×50` точек (пикселей) – кружок **красного** цвета – и сохраните его в формате `.gif` в файле под именем `img.gif`.  
Для этого, например:
  - В операционной системе Linux:
    - а) запустите на выполнение графический редактор GIMP, выполнив команду **Меню ⇒ Графика ⇒ Редактор растровой графики GIMP**;
    - б) в окне **Редактор растровой графики GIMP** выполните команду **Файл ⇒ Создать**; Если некоторые элементы управления невидны – перетащите окно за заголовок по экрану монитора так, чтобы они стали видны.

- в) в раскрывшемся окне **Создать новое изображение** укажите **Размер изображения 50 x 50 точек раstra** и щелкните по кнопке **ОК**;
- г) в окне ... **50 x 50 – GIMP** выполните команду **Файл ⇨ Сохранить**;
- д) в раскрывшемся окне **Сохранить изображение** укажите:
- в поле **Имя** – `img.gif`;
  - при помощи раскрывающегося списка **Просмотреть другие папки** – `lr3`;
  - при помощи раскрывающегося списка **Выберите тип файла** – **GIF image** и
  - щелкните по кнопке **Сохранить**;
- е) в раскрывшемся далее окне **Экспортировать файл** выберите значение переключателя по умолчанию – **Преобразовать в индексированное...** – и щелкните по кнопке **Экспорт**;
- ж) в раскрывшемся далее окне **Сохранить как GIF** щелкните по кнопке **Сохранить**;
- з) в окне **Панель инструментов** выберите инструмент **Эллиптическое выделение**;
- и) при нажатой клавише **Shift** в поле рисования выделите окружность, перетаскив указатель мыши по диагонали из одного угла в другой;
-  *При перемещении по рисунку любого инструмента его точные координаты отображаются внизу окна – в строке состояния.*
- к) в окне **Панель инструментов** щелкните по инструменту **Цвет переднего плана**;
- л) в раскрывшемся далее окне **Цвет переднего плана** в поле редактирования **HTML-разметка** введите код **красного** цвета – «`ff0000`» – и щелкните по кнопке **ОК**;
- в результате цвет переднего плана изменится на **красный**.
- м) в окне **Панель инструментов** щелкните по инструменту **Плоская заливка** ;
- н) щелкните левой кнопкой мыши по выделенной области в поле рисования;
- в результате выделенная область – круг – будет закрашена в **красный** цвет.
- о) выполните команду **Файл ⇨ Сохранить** и подтвердите необходимость экспортирования (преобразования) изображения;
- п) выполните команду **Файл ⇨ Выход**;

- В операционной системе Windows:
  - а) запустите на выполнение графический редактор Paint, например, выполнив команду **Пуск ⇒ Все программы ⇒ Стандартные ⇒ Paint**;
  - б) в раскрывшемся окне **Безымянный - Paint** выполните команду **Рисунок ⇒ Атрибуты...**;
  - в) в раскрывшемся далее окне диалога **Атрибуты** укажите в полях ввода **Ширина** и **Высота** – 50 точек и щелкните по кнопке **ОК**;
  - г) в окне **Безымянный - Paint** выполните команду **Файл ⇒ Сохранить**;
  - д) в раскрывшемся окне диалога **Сохранить как** укажите:
    - в поле редактирования **Папка** – lr3;
    - в поле редактирования **Имя файла** – img.gif;
    - в раскрывающемся списке **Тип файла** – GIF (\*.GIF);
    - щелкните по кнопке **Сохранить**;
  - е) в раскрывшемся окне диалога подтвердите выполнение сохранения, щелкнув по кнопке **ДА**;
  - ж) в окне **img.GIF - Paint**:
    - 1) выполните команду **Вид ⇒ Масштаб ⇒ Крупный**;
    - 2) щелкните левой кнопкой мыши по **красному** цвету на палитре цветов внизу окна;
    - 3) щелкните левой кнопкой мыши по инструменту рисования **Эллипс**;
    - 4) укажите сплошную заливку без контура, щелкнув по последнему прямоугольнику ниже инструмента рисования **Эллипс**;
    - 5) при нажатой клавише **Shift** нарисуйте в поле рисования кружок **красного** цвета, перетаскив указатель мыши по диагонали из одного угла в другой;

 *При перемещении по рисунку любого инструмента его точные координаты отображаются внизу окна – в строке состояния.*
  - з) выполните команду **Файл ⇒ Сохранить**;
  - и) выполните команду **Файл ⇒ Выход**;

2. Отредактируйте файл index.html, поместив после строки:
 

«Приветствую Вас на своей страничке!<br><br>»

тег вставки созданного рисунка img.gif:

«<br><br>».



3. В браузере просмотрите файл `index.html`.

Если в окне браузера вместо картинки отображается стилизованный квадрат, то это значит, что:

- рисунок создан не в том формате или
- неправильно указано его расположение.

Для исправления ошибки необходимо еще раз внимательно выполнить все пункты текущего **Шага**.

## **Шаг 4 Использование рисунков в качестве гиперссылок**

1. Строку файла `index.html`:

```
«<br><br>»
```

отредактируйте следующим образом:

```
«<a href="hello/hello.html" title=" Перейти к  
приветствию"></a><br><br>».
```

2. Обновите файл `index.html` в окне браузера.

Обратите внимание, что в некоторых браузерах вокруг рисунка появляется рамка. Так происходит потому, что рисунок используется в качестве гиперссылки. Для того чтобы убрать рамку необходимо явно указать ее нулевую ширину с помощью атрибута `border="0"` тега `<img>`.

## **Шаг 5 Создание гиперссылок на объекты внутри документа**

1. В файл `index.html` внесите такие изменения:

- а) строку «Приветствую Вас на своей страничке!» пометьте при помощи атрибута `name="top"` тега `<a>`,
- б) при помощи тега `<br>` в конце тела страницы, перед закрывающимся тегом `</body>`, вставьте 5 пустых строк,
- в) перед закрывающимся тегом `</body>` вставьте ссылку на начало документа,
- г) саму ссылку на начало страницы пометьте при помощи атрибута `id="bottom"`.

После внесенных изменений файл должен выглядеть следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Главная страница</title>
  </head>
  <body>
    <a name="top">Приветствую Вас на своей
страничке!</a><br><br>
    <a href="hello/hello.html" title="Перейти к
приветствию"></a><br><br>
    Вот мое <a href="hello/hello.html">приветствие</a>
    <br>
    <br>
    <br>
    <br>
    <br>
    <a id="bottom" href="#top">Перейти в начало
документа</a>
  </body>
</html>
```

2. Обновите файл `index.html` в окне браузера.
3. Уменьшите размер окна браузера по высоте на столько, чтобы файл `index.html` в него полностью не помещался.
4. При помощи вертикальной полосы прокрутки покажите окончание документа.
5. Проверьте работу ссылки на начало документа.
6. Отредактируйте файл `hello.html`, внося в него, перед закрывающимся тегом `</p>`, гиперссылку на метку «`bottom`» в файле `index.html`.

В результате файл `hello.html` будет иметь такое содержание:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Мой первый документ HTML</title>
  </head>
  <body>
    <p align="center">
      <font color="#008080" size="7"><b>Всем</b></font><br>
      <font size="6"><i>огромный привет!</i></font><br>
      <a href="../index.html">Вернуться к главной
странице</a><br>
      <a href="../index.html#bottom">В конец главной
страницы</a>
    </p>
  </body>
</html>
```

7. Проверьте работу всех ссылок.

## Шаг 6 Создание внешних гиперссылок

1. В файле `index.html` создайте гиперссылку на сайт Университета.

В результате он должен иметь такое содержание:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Главная страница</title>
  </head>
  <body>
    <a name="top">Приветствую Вас на своей
страничке!</a><br><br>
    <a href="hello/hello.html" title="Перейти к
приветствию"></a>
    <br><br>
    Вот мое <a href="hello/hello.html">приветствие</a>
    <br>
    <br>
```

```

Здесь я учусь
<br>
<br>
<br>
Перейти в начало
документа
</body>
</html>

```

## 2. Проверьте работу ссылки.

Если Ваш компьютер подключен к Интернету, будет открыта стартовая страница сайта Университета, если нет – выдана ошибка о том, что попытка соединения не удалась и запрашиваемую станицу отобразить невозможно.

## Шаг 7 Создание гиперссылок на адреса электронной почты

### 1. В файле index.html создайте гиперссылку на произвольный адрес электронной почты или на свой – если он у Вас есть.

В результате он должен иметь такое содержание:

```

<html>
<head>
  <meta charset=utf-8>
  <title>Главная страница</title>
</head>
<body>
  <a name="top">Приветствую Вас на своей
страничке!</a><br><br>
  <a href="hello/hello.html" title="Перейти к
приветствию"></a>
  <br><br>
  Вот мое <a href="hello/hello.html">приветствие</a>
  <br>
  <br>
  <a href="http://www.kname.edu.ua/">Здесь я учусь</a>
  <br>
  <br>

```

```
<a href="mailto:borisu-p@ukr.net">Мой e-mail</a>
<br>
<br>
<a id="bottom" href="#top">Перейти в начало
документа</a>
</body>
</html>
```

## 2. Проверьте работу ссылки.

Должна запускаться установленная по умолчанию в Вашей системе программа-клиент работы с электронной почтой.

### **Индивидуальные задания**

1. В созданном ранее каталоге lr3 создайте каталог iz и сделайте его текущим.
2. Создайте файл index.html, так чтобы:
  - в заголовке окна была фраза «Это я»;
  - Вашу Фамилию Имя и Отчество отформатировать заголовком 3-го уровня;
  - содержал графическое изображение или Ваше фото;
  - информация, например, о:
    - дате рождения,
    - месте проживания,
    - родителях,
    - любимых предметах,
    - увлечениях
    - и т. д.была вынесена в отдельные файлы, а связь с основным выполнялась при помощи гиперссылок;
  - добавьте так же гиперссылки на сайт Университета и адрес Вашей электронной почты.
3. В тетрадь для выполнения лабораторных работ занесите эскизы созданных Web-страниц и, соответствующие им, тексты HTML-документов.

## ***Контрольные вопросы***

1. Что такое гипертекстовый документ?
2. Что такое гиперссылка?
3. Что такое внутренняя ссылка?
4. Что такое внешняя ссылка?
5. Как создать гиперссылку на определенное место внутри другого HTML-документа?
6. Какие есть способы создания меток в HTML-документе для создания на них гиперссылок?
7. Что такое относительная ссылка и как она создается?
8. Как указать гиперссылку на файл в текущем каталоге?
9. Как указать гиперссылку на файл в родительском каталоге (уровнем выше)?
10. Как указать гиперссылку на файл во вложенном каталоге (уровнем ниже)?
11. Какой документ откроет браузер, если в гиперссылке указан только сайт (каталог), без указания конкретного документа?
12. Как указать цвет гиперссылки?
13. Как указать цвет выделенной гиперссылки?
14. Как указать цвет уже посещаемой гиперссылки?
15. Как указать загрузку гиперссылки в текущее окно браузера?
16. Как указать загрузку гиперссылки в новое окно браузера?
17. Как указать подсказку, когда в качестве гиперссылки используется изображение?
18. Как создать гиперссылку на адрес электронной почты?
19. Как создать гиперссылку на news-конференцию?
20. Какие форматы файлов изображений можно вставлять в HTML-документ?
21. Как создать графическое изображение в операционной системе Linux?
22. Как создать графическое изображение в операционной системе Windows?
23. Как вставить графическое изображение в HTML-документ?
24. Как указывается имя файла, содержащего графическое изображение?
25. Как выровнять изображение полевому краю окна браузера?
26. Как выровнять изображение правому краю окна браузера?
27. Как установить, или наоборот – убрать, рамку вокруг изображения?
28. Как установить размер свободного места вокруг изображения?
29. Как указать размер изображения?
30. Как использовать изображение в качестве гиперссылки?
31. Сформулируйте свой вопрос по пройденному материалу.
32. Сформулируйте свое предложение по совершенствованию темы.

## Лабораторная работа № 4

### Таблицы

#### *Цель работы*

Освоение и приобретение практических навыков работы с таблицами в HTML-документах.

#### ***Краткие теоретические сведения***

Таблицы в HTML-документе создаются за четыре этапа:

- **На первом этапе** в HTML-коде с помощью парного тега `<table>` формируют саму таблицу:  
`<table>`  
`</table>`.
- **На втором этапе** формируют строки таблицы. Для этого предусмотрены парные теги `<tr>` (Table Row); каждый такой тег создает отдельную строку. Теги `<tr>` помещают внутрь тега `<table>`.
- **На третьем этапе** создают ячейки таблицы, для чего используют парные теги `<td>` (Table Data) и `<th>` (Table Header). Тег `<td>` создает обычную ячейку, а тег `<th>` – ячейку заголовка, в которой будет помещаться «шапка» соответствующего столбца таблицы. Теги `<td>` и `<th>` помещают в теги `<tr>`, создающие строки таблицы, в которых должны находиться эти ячейки.
- **На четвертом, последнем, этапе** указывают содержимое ячеек, которое помещают в соответствующие теги `<td>` и `<th>`.

Ячейки таблицы HTML должны иметь хоть какое-то содержимое, иначе Web-обозреватель может их вообще не отобразить. Если же какая-то ячейка должна быть пустой, в нее следует поместить неотображаемый символ «неразрывный пробел» (`&nbsp;`). Если одна или несколько ячеек, располагающихся в конце какой-либо строки, не содержат данных, то их описание может быть опущено, а браузер автоматически добавит требуемое количество пустых ячеек. Отсюда следует, что построение таблиц, в которых в разных строчках располагается различное количество столбцов одного и того же размера, не разрешается.

По умолчанию размер таблицы устанавливается в зависимости от ее содержания так, чтобы она отображалась наилучшим образом. С помощью атрибутов `width=` и `height=` можно явно указать размеры ячейки, строки и всей таблицы. Их значения могут быть указаны либо в пикселях, либо в

процентах. Горизонтальное выравнивание в таблицах устанавливается атрибутом `align=`, а вертикальное – `valign=`.

Различие в использовании тегов `<td>` и `<th>` заключается лишь в типе шрифта, используемого по умолчанию для отображения содержимого ячеек, а также расположению данных внутри ячейки. Содержимое ячеек типа `<th>` отображается полужирным (**Bold**) шрифтом и располагается по центру (`align="center"`, `valign="middle"`). Ячейки, определенные тегом `<td>` по умолчанию отображают данные, выровненные влево (`align="left"`) и посередине (`valign="middle"`) в вертикальном направлении.

Завершающие теги `</td>`, `</th>` и `</tr>` могут быть опущены. В этом случае концом описания строки или ячейки является начало следующей строки или ячейки, или конец таблицы. Завершающий же тег таблицы `</table>` не может быть опущен.

Таблица может иметь заголовок, который заключается в пару тегов `<caption>...</caption>`. Описание заголовка таблицы должно располагаться внутри тегов `<table>...</table>` до первого `<tr>`. По умолчанию текст заголовка таблицы располагается над ней – `align="top"` – и центрируется по горизонтали. Можно так же указать расположение заголовка под таблицей – `align="bottom"`.

В таблицах HTML-документов смежные ячейки можно объединять в одну, как по горизонтали, так и по вертикали. Специально для этого теги `<td>` и `<th>` поддерживают два необязательных атрибута. Первый – `colspan=` – объединяет ячейки по горизонтали, второй – `rowspan=` – по вертикали.

Чтобы объединить несколько ячеек по горизонтали в одну, нужно выполнить следующие шаги:

1. Найти в коде HTML тег `<td>` (`<th>`), соответствующий первой из объединяемых ячеек (если считать ячейки слева направо).
2. Вписать в него атрибут `colspan=` и присвоить ему количество объединяемых ячеек, считая и самую первую из них.
3. Удалить теги `<td>` (`<th>`), создающие остальные объединяемые ячейки данной строки.

Объединить ячейки по вертикали чуть сложнее. Вот шаги, которые нужно для этого выполнить:

1. Найти в коде HTML строку (тег `<tr>`), в которой находится первая из объединяемых ячеек (если считать строки сверху вниз).
2. Найти в коде этой строки тег `<td>` (`<th>`), соответствующий первой из объединяемых ячеек.
3. Вписать в него атрибут `rowspan=` и присвоить ему количество объединяемых ячеек, считая и самую первую из них.



4. Просмотреть последующие строки и удалить из них теги <td> (<th>), создающие остальные объединяемые ячейки.

HTML-код, создающий таблицы, может показаться несколько громоздким. Но это плата за исключительную гибкость таблиц HTML. В них можно поместить любое содержимое, например: абзацы, заголовки, изображения, аудио- и видеоролики и даже другие таблицы.

Таблицы в HTML используются не только традиционно, как метод представления данных, но и как средство разметки Web-страниц. Так плавающие, или встроенные, фреймы, которые создаются при помощи тега <iframe>, позволяет вставлять один HTML-документ в другой наподобие матрешек. В частности можно «загружать» («подгружать») в ячейки таблицы необходимые HTML-документы. Если браузер пользователя не поддерживает плавающих фреймов, то соответствующее сообщение необходимо поместить внутри контейнера <iframe>...</iframe>. Размеры плавающего фрейма указываются в атрибутах width= и height=, а в frameborder= – отображать (1) или не отображать (0) рамку вокруг него.

Все, с теорией закончили. Настала пора практики. В конце концов, учиться лучше всего на примерах. И дальше – по инструкции:

### ***Ход работы***

#### **Шаг 1 Создание простейшей таблицы 2x3**

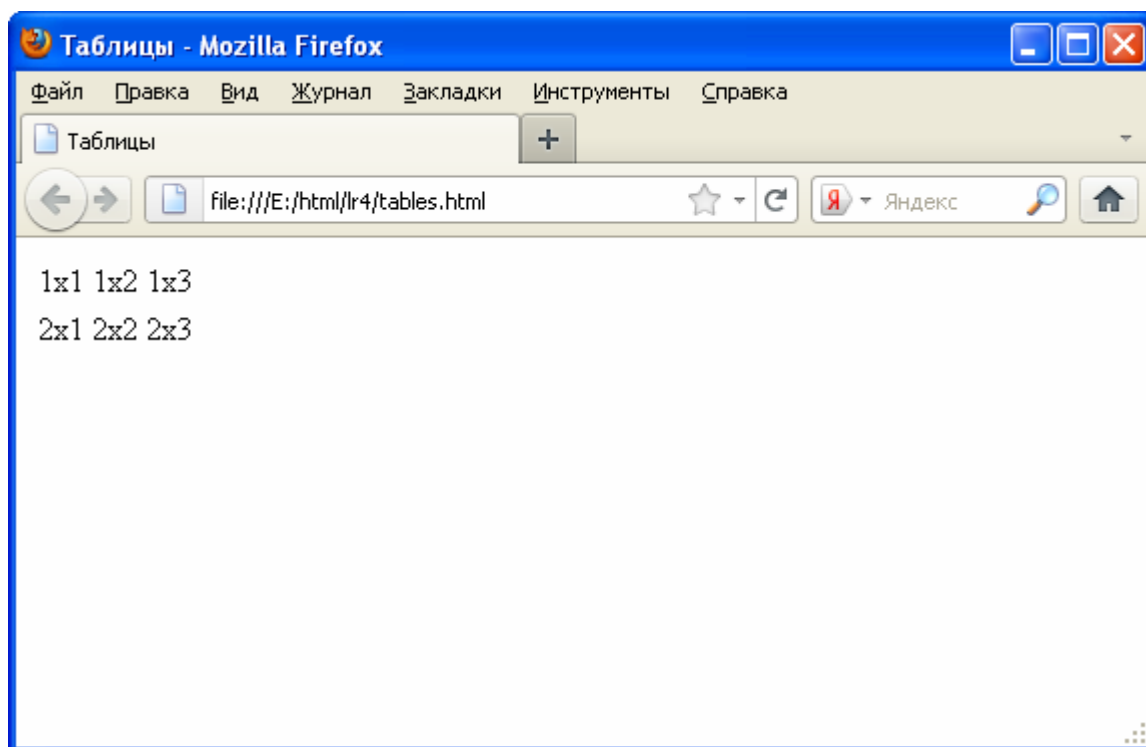
1. В каталоге для выполнения лабораторных работ по HTML – html, создайте каталог lr4 и сделайте его текущим.
2. На основе созданного в первой Лабораторной работе шаблона Web-страницы template.html, или создайте заново, файл tables.html, который содержит простейшую таблицу, состоящую из 2-х строк и 3-х столбиков, в каждой ячейке которой записаны ее «координаты», например, «1x1» для – 1-й ячейки 1-й строки, «1x2» – для 2-й ячейки 1-й строки, и т. д.:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Таблицы</title>
  </head>
  <body>
    <table>
      <tr>
```

```
<td>1x1</td><td>1x2</td><td>1x3</td>
</tr>
<tr>
  <td>2x1</td><td>2x2</td><td>2x3</td>
</tr>
</table>
</body>
</html>
```

3. Просмотрите созданную таблицу в браузере.

Если не было допущено ошибок, то она должна выглядеть примерно так:



Столь незатейливый вид таблицы обусловлен тем, что так выглядят все таблицы с параметрами тегов, принятыми по умолчанию.

## Шаг 2 Добавление рамок

1. В файле `tables.html` отредактируйте тег создания таблицы `<table>`, добавив к нему атрибут управления рамкой `border=`, указав ее толщину в 1 пиксель:  
«`<table border="1">`».
2. Сохраните отредактированный файл `tables.html`, обновите содержимое окна браузера, и Вы увидите результат.

### Шаг 3 Пустые ячейки

1. Отредактируйте файл `tables.html`, «пропустив» в нем 2-ю ячейку в 1-й строке, т. е. тег описания 1-й строки будет выглядеть следующим образом:  
«`<td>1x1</td><td>1x3</td>`».

2. Посмотрите полученный результат.  
Он должен выглядеть примерно так:

1x1	1x3	
2x1	2x2	2x3

3. Снова отредактируйте файл `tables.html`, вставив в нужном месте «пустую» ячейку 1x2.

В результате тег описания 1-й строки должен выглядеть следующим образом:

«`<td>1x1</td><td></td><td>1x3</td>`».

4. Посмотрите изменения в браузере.

В результате место под ячейку 1x2 будет выделено, но рамки вокруг нее не будет.

5. Для того, чтобы вокруг пустой ячейки отображалась рамка, она должна содержать, по крайней мере, один из неотображаемых символов. Например, «неразрывный пробел» – «`&nbsp;`». В результате описание 1-й строки должно выглядеть следующим образом:

«`<td>1x1</td><td>&nbsp;</td><td>1x3</td>`».

### Шаг 4 Объединение ячеек

1. Отредактируйте файл `tables.html` так, чтобы 1-я и 2-я ячейки 1-го ряда были объединены.

Для этого необходимо отредактировать описание 1-й строки следующим образом:

«`<td colspan="2">1x1</td><td>1x2</td>`».

2. Посмотрите полученный результат в браузере.

Он должен выглядеть примерно так:

1x1	1x2	
2x1	2x2	2x3

3. Отредактируйте описание 1-й строки следующим образом – т. е. «забудьте» удалить ячейку 1x3:

«<td colspan="2">1x1</td><td>1x2</td><td>1x3</td>».

4. Посмотрите полученный результат.

Он должен выглядеть примерно так:

1x1	1x2	1x3
2x1	2x2	2x3

5. Самостоятельно, при помощи атрибута `rowspan=`, объедините ячейки 1x3 и 2x3 так, чтобы получить такую таблицу:

1x1	1x2	1x3
2x1	2x2	2x3

### Шаг 5 Вложенные таблицы

1. Создайте таблицу, состоящую из 1-й строки и 2-х столбиков. Во второй столбик поместите таблицу, состоящую из 2-х строк и 1-го столбика.

Для этого отредактируйте файл `tables.html` следующим образом:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Таблицы</title>
  </head>
  <body>
    <table border="1">
      <tr>
        <td>1+2</td>
        <td>
          <table border="1">
            <tr><td>1x2</td></tr>
            <tr><td>2x2</td></tr>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```

2. Посмотрите полученный результат.

Он должен выглядеть примерно так:

1+2	1x2
	2x2

Вложенные таблицы и объединение ячеек – два способа решения одной задачи. Какой из них предпочесть, зависит о решаемой задачи. Объединение ячеек – более простой метод, и применяется в более простых ситуациях; вложенные таблицы позволяют более гибко конструировать сложные таблицы.

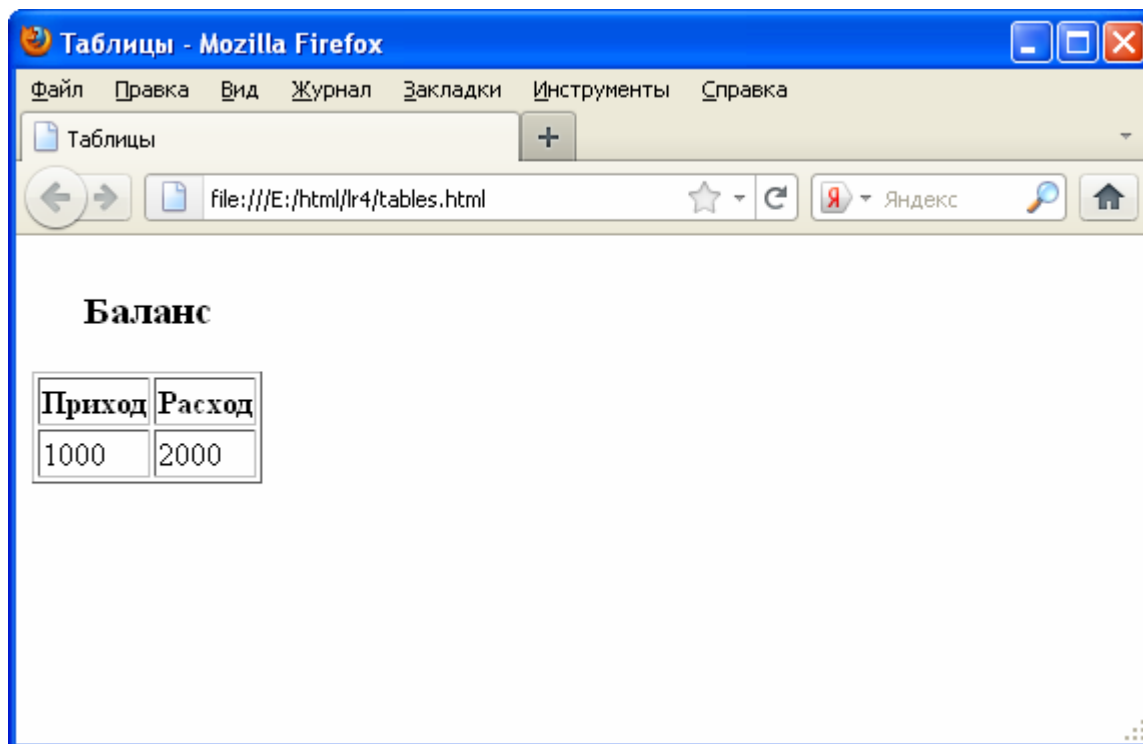
## Шаг 6 Добавление заголовков

1. Отредактируйте файл `tables.html`, добавив к таблице заголовок «Баланс», а 1-му и 2-му столбикам – «Приход» и «Расход», соответственно.

В результате он должен иметь такое содержание:

```
<html>
<head>
  <meta charset=utf-8>
  <title>Таблицы</title>
</head>
<body>
  <table border="1">
    <caption><h3>Баланс</h3></caption>
    <tr>
      <th>Приход</th><th>Расход</th>
    </tr>
    <tr>
      <td>1000</td><td>2000</td>
    </tr>
  </table>
</body>
</html>
```

2. Посмотрите полученный результат в браузере.  
Он должен выглядеть примерно так:



Заголовок таблицы добавляется при помощи тега `<caption>...</caption>`. По умолчанию он отображается основным текстом таблицы. Для того, чтобы его выделить был использован тег заголовка `<h3>...</h3>`. Заголовки столбиков выводятся при помощи тега `<th>...</th>`. По умолчанию он выводится полужирным начертанием, выровненным по середине ячейки. Содержимое ячеек созданных при помощи тега `<td>...</td>` выводится нормальным шрифтом, выровненным по левому краю ячейки. Обратите также внимание, что размер ячеек был подобран под их наибольшее содержание.

### Шаг 7 Использование таблиц для разметки Web-страниц

Создайте миниWeb-сайт для отображения личных страниц Лизы, Иры, Кати, Светы и Наташи. Он должен иметь такую структуру:

Меню	Логотип
	Содержание

В левом поле располагается меню, при помощи которого в правое нижнее поле будут загружаться персональные страницы. В правом верхнем поле должен отображаться логотип.

1. Создайте Web-страницу `liza.html`, с информацией о Лизе, например, такого содержания:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Лиза</title>
  </head>
  <body>
    Меня зовут Лиза
  </body>
</html>
```

2. Создайте аналогичные HTML-документы для следующих 4-х женских имен: Ира, Катя, Света и Наташа, соответственно, `ira.html`, `katya.html`, `sveta.html` и `natasha.html`.
3. Создайте файл `menu.html`, содержащий гиперссылки на страницы HTML с информацией о Лизе, Ире, Кате, Свете и Наташе:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Меню</title>
  </head>
  <body>
    <h3>Меню</h3>
    <a href="liza.html" target="content">Лиза</a><br>
    <a href="ira.html" target="content">Ира</a><br>
    <a href="katya.html" target="content">Катя</a><br>
    <a href="sveta.html" target="content">Света</a><br>
    <a href="natasha.html" target="content">Наташа</a>
  </body>
</html>
```

4. При помощи графического редактора создайте логотип размером 80×80 точек (пикселей), например, кружок **зеленого** цвета, и сохраните его в формате `.gif` в файле под именем `logo.gif`.

5. Отредактируйте файл `tables.html` так, чтобы он имел следующее содержание:

```
<html>
  <head>
    <meta charset=utf-8>
    <title>Таблицы</title>
  </head>
  <body>
    <table height="100%">
      <tr valign="top">
        <td>
          <iframe src="menu.html" width="80"
height="100%" frameborder="0">
            <font color="ff0000">Чтобы увидеть эту
страницу обновите браузер</font>
          </iframe>
        </td>
        <td>
          <table>
            <tr><td>
              
            </td></tr>
            <tr><td>
              <iframe src="liza.html" name="content"
frameborder="0"></iframe>
            </td></tr>
          </table>
        </td>
      </tr>
    </table>
  </body>
</html>
```

6. Проверьте работу созданного миниWeb-сайта.
7. Самостоятельно удалите, или измените, значение атрибутов `width=`, `height=` и `valign=`, которые служат для явной установки ширины, высоты и вертикального выравнивания, соответственно, и посмотрите, как «поплывет» таблица при изменении размеров окна браузера.



## **Индивидуальные задания**

1. Создайте каталог `iz` и сделайте его текущим.
2. Создайте Web-страницу `result.html`, которая содержит таблицу с результатами предыдущей сессии. Каждая строка таблицы должна иметь такие колонки (как в зачетной книжке):
  - Номер по порядку,
  - Наименование учебной дисциплины,
  - Объем часов по учебному плану,
  - Фамилия Имя Отчество преподавателя,
  - Дата проведения,
  - Экзаменационная оценка или информацию о зачете.
3. Создайте личный миниWeb-сайт, на котором будете отображать информацию о:
  - себе,
  - родителях,
  - друзьях,
  - любимых предметах,
  - увлечениях,
  - результатах предыдущей сессии (файл `result.html`)
  - и т. д.Разметка главной страница должна быть выполнена с помощью таблицы. Она должна содержать минимум 3 области: меню, логотип и содержание.
4. В тетрадь для выполнения лабораторных работ занесите эскизы созданных Web-страниц и, соответствующие им, тексты HTML-документов.

## **Контрольные вопросы**

1. Как создать таблицу в HTML-документе?
2. Как должна начинаться и заканчиваться таблица в HTML-документе?
3. Из чего состоит таблица?
4. Как начинаются и завершаются строки таблицы?
5. Как создаются ячейки в строке?
6. Для чего служит тег `<td>`?
7. Для чего служит тег `<th>`?
8. В чем отличие тега `<th>` от `<td>`?

9. Где могут появляться, а где нет, теги `<td>` и `<th>`?
10. Обязательно ли наличие тегов `</td>`, `</th>` и `</tr>`?
11. Обязательно ли наличие тега `</table>`?
12. Как определяется количество строк в таблице?
13. Как определяется количество столбцов в таблице?
14. Как создаются пустые ячейки внутри строки?
15. Как создаются пустые ячейки в конце строки?
16. Можно ли создать таблицу, в строках которой расположено различное количество столбцов одинаковой ширины?
17. Как создается заголовок таблицы?
18. Где может располагаться описание заголовка таблицы?
19. Где по умолчанию располагается заголовок таблицы?
20. Где может располагаться заголовок таблицы?
21. Как устанавливается размер таблицы?
22. Какие атрибуты явно задают размер таблицы?
23. В каких единицах устанавливаются размеры таблицы?
24. Как явно установит горизонтальное и вертикальное выравнивание в таблице?
25. Как объединять столбцы таблицы?
26. Как объединять строки таблицы?
27. Допускается ли вложенность таблиц в документах HTML?
28. Как используются таблицы в HTML-документах?
29. Для чего служит тег `<iframe>`, какие он имеет атрибуты?
30. Как предотвратить «плавание» таблицы при изменении размеров окна браузера?
31. Сформулируйте свой вопрос по пройденному материалу.
32. Сформулируйте свое предложение по совершенствованию темы.

## Список источников

1. Дронов В. А. HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. – СПб.: БХВ-Петербург, 2011. – 416 с.: ил.
2. Коржинский С. Н. Настольная книга Web-мастера: эффективное применение HTML, CSS, JavaScript. Издание второе, исправленное и дополненное. – М.: Издательский торговый дом «КноРус», 2000. – 320 с.
3. Матросов А. А., Сергеев А. О., Чаунин М. П. HTML 4.0. – БХВ-Петербург, 2004. – 672 с.: ил.
4. Алленова Наталья Учебник по Html (хтмл) для чайников. – [Электронный ресурс]. – Режим доступа: <http://postroika.ru/html/>.
5. Городулин Владимир HTML-справочник. – [Электронный ресурс]. – Режим доступа: <http://html.manual.ru/>.
6. Климов Александр HTML в примерах. – [Электронный ресурс]. – Режим доступа: <http://webmaster.alexanderklimov.ru>.

*Навчальне видання*

Методичні вказівки

для виконання лабораторних, практичних,  
самостійних і контрольних робіт  
з курсу

## **«ІНФОРМАТИКА»**

*(для студентів 1-го курсу денної та заочної форм навчання  
напрямів підготовки 6.030504 – Економіка підприємств  
та 6.030509 – Облік та аудит)*

*(рос. мовою)*

Укладач **ПОГРЕБНЯК** Борис Іванович  
**ШТЕЛЬМА** Ольга Миколаївна

Відповідальний за випуск *О. Б. Костенко*

За авторською редакцією

Комп'ютерне верстання *І. В. Волосожарова*

План 2015, поз. 389М

Підп. до друку 25.03.2015р.	Формат 60х84/16
Друк на ризографі	Ум. друк. арк. 3,5
Тираж 50 пр.	Зам. №

Видавець і виготовлювач:

Харківський національний університет  
міського господарства імені О. М. Бекетова,  
вул. Революції, 12, Харків, 61002  
Електронна адреса: rectorat@kname.edu.ua  
Свідоцтво суб'єкта видавничої справи:  
ДК № 4705 від 28.03.2014р.