

БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ

Кафедра «Автоматизированные системы управления»

ТЕХНОЛОГИЯ ИНТЕРНЕТ- ПРОГРАММИРОВАНИЯ

*Методические указания к лабораторной
работе № 1 для студентов специальности*
**09.03.01 Автоматизированные системы обработки
информации и управления**

Могилев 2020

Лабораторная работа №1
Создание HTML страницы с применением CSS

Цель работы: Создание HTML страницы с применением CSS

Порядок выполнения работы.

Изучить теоретические сведения.

Выполнить задание к лабораторной работе в соответствии с вариантом.

Оформить отчет.

Требования к отчету.

Цель работы.

Постановка задачи.

Текст программы.

ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1 ОСНОВНЫЕ ПОНЯТИЯ HTML

HTML – язык гипертекстовой разметки документов (HyperText Markup Language). С помощью HTML создаются Web-страницы, которые находятся в глобальной компьютерной сети Интернет. HTML – это не язык программирования в традиционном смысле, он является языком разметки. С помощью HTML текстовый документ разбивают на блоки смысловой информации (заголовки, параграфы, таблицы, рисунки и т.п.).

Гипертекстовый документ – это документ, содержащий переходы (**гиперссылки**) на другие документы. При использовании гиперссылки происходит перемещение от одного документа к другому (как по цепочке) в Интернете. HTML-документ является гипертекстовым документом.

Особенности HTML-документа:

1. HTML-документ может содержать текст, графику, видео и звук.
2. В общем случае HTML-документ – это один или несколько текстовых файлов, имеющих расширение .htm или .html.
3. Создавать HTML-документ можно как с помощью специальных программ – редакторов HTML (например, FrontPage), так и с помощью любого текстового редактора (например, блокнота Windows).
4. Для просмотра HTML-документов существуют специальные программы-**браузеры**. Они **интерпретируют** HTML-документы, т.е. переводят текст документа в Web-страницу, и отображают ее на экране пользователя. Существует очень много различных браузеров, но наиболее распространенными браузерами являются Microsoft Internet Explorer, Netscape Navigator и Opera.

5. Если при интерпретации HTML-документа браузер чего-то не понимает, то сообщения об ошибке не возникает, а это место в HTML-документе игнорируется и не отображается браузером.

HTML-документ состоит из элементов HTML.

Элемент HTML – это чаще всего два тега (открывающий и закрывающий) и часть документа между ними. Кроме того, существуют элементы HTML, состоящие только из одного тега.

Тег – в переводе с английского – ярлык, этикетка. Тег определяет тип выводимого элемента HTML (например, заголовок, таблица, рисунок и т.п.). Сам тег не отображается браузером. Тег представляет собой последовательность элементов:

- символ левой угловой скобки (<) – начало тега;
- необязательный символ слеша (/) – символ используется, чтобы обозначить закрывающий тег;
- имя тега;
- необязательные атрибуты в открывающем теге;
- символ правой угловой скобки (>)

Атрибуты – необязательный набор параметров, определяющих дополнительные свойства элемента HTML (например, цвет или размер). Атрибут состоит:

- из имени атрибута;
- знака равенства (=);
- значения атрибута – строки символов, заключенной в кавычки

Пример элемента HTML:

```
<H1 ALIGN= "CENTER">Глава 1</H1>
```

В этом примере:

<H1 ALIGN= "CENTER"> – открывающий тег

</H1> – закрывающий тег

H1 – имя тега

ALIGN= "CENTER" – атрибут

ALIGN – имя атрибута

"CENTER" – значение атрибута

Глава 1 – содержание элемента HTML

Правила создания HTML-документов:

1. Теги и атрибуты можно записывать в любом регистре, т.е. </H1> и </h1> – это одно и то же.

2. Несколько пробелов подряд, символы табуляции и перевода строки при интерпретации браузером заменяются на один пробел. Это позволяет писать хорошо структурированные исходные тексты файлов HTML.

3. Рекомендуются давать имена файлам HTML строчными английскими буквами. Длина имени – до восьми символов. В принципе, можно не

придерживаться данной рекомендации, но тогда пользователи, работающие в операционных системах, отличных от Windows, не смогут воспользоваться вашими HTML-документами.

2. СТРУКТУРА HTML-ДОКУМЕНТА

Каждый HTML-документ должен начинаться тегом **<HTML>** и заканчиваться тегом **</HTML>**. Эти теги обозначают, что находящиеся между ними строки представляют единый HTML-документ. Кроме того, можно заметить, что файл HTML в целом является элементом языка HTML.

Также в HTML-документе должны присутствовать элементы **HEAD** (информация о документе) и **BODY** (тело документа).

2.1. РАЗДЕЛ ДОКУМЕНТА HEAD

Раздел документа **HEAD** определяет его заголовок, а также содержит дополнительную информацию о документе для браузера. Этот раздел не является обязательным, однако рекомендуется всегда использовать его в своих HTML-документах, так как правильно составленный заголовок может быть весьма полезен.

Раздел заголовка начинается тегом **<HEAD>** и следует сразу за тегом **<HTML>**. Между открывающим и закрывающим тегами элемента **HEAD** располагаются другие элементы заголовка.

Название документа TITLE

Для того чтобы дать название HTML-документу, предназначен тег **<TITLE>**. Это название будет выведено в заголовок окна браузера. Название записывается между тегами **<TITLE>** и **</TITLE>** и представляет собой строку текста. Длина этой строки может быть любой, но рекомендуется делать ее не больше 60 символов. Элемент **TITLE** должен находиться только в разделе **HEAD**.

2.2. РАЗДЕЛ ДОКУМЕНТА BODY

В этом разделе документа располагается та информация, которая отображается в окне браузера. Раздел **BODY** должен начинаться тегом **<BODY>** и завершаться тегом **</BODY>**, между которыми располагаются элементы HTML, из которых и состоит содержание документа.

2.2.1. Спецификация элемента BODY

Тег **<BODY>** имеет ряд атрибутов, определяющих внешний вид документа. Ниже приводится спецификация тега **<BODY>**.

<BODY

TEXT="цвет текста"
BGCOLOR="цвет фона"
BACKGROUND="адрес фонового рисунка"
LINK="цвет непосещенной гиперссылки"
VLINK="цвет посещенной гиперссылки"
ALINK="цвет активной гиперссылки"

>

Атрибут TEXT задает цвет шрифта для всего документа в формате RGB или в символьной нотации. По умолчанию (если не указан этот атрибут) используются настройки броузера.

Атрибут BGCOLOR задает цвет фона окна броузера документа в формате RGB или в символьной нотации. По умолчанию используются настройки броузера.

Атрибут BACKGROUND позволяет указать адрес и имя рисунка, используемого в качестве фона. Этот рисунок будет размножен и распределен на заднем плане документа.

Атрибуты LINK, VLINK и ALINK задают цвета гиперссылок в формате RGB или в символьной нотации. По умолчанию используются настройки броузера. Непосещенная гиперссылка – гиперссылка, которая еще не использовалась для перехода к другому документу. Посещенная гиперссылка – гиперссылка, которая уже использовалась для перехода к другому документу. Активная гиперссылка – гиперссылка на документ, к которому в данный момент происходит переход.

2.2.2. Советы по использованию атрибутов тега BODY

- Если вы указываете хотя бы один цвет в теге BODY, то укажите и остальные. Это связано с тем, что пользователь может установить настройки цветов своего броузера как ему удобней. Указание только одного цвета может привести к ситуации, что у некоторых пользователей текст сольется с цветом фона. В результате просмотр документа будет затруднен.

- Выбирайте цвет текста так, чтобы он "работал" вместе с цветом фона или основными цветами изображения. Например, красное на зеленом может вызвать серьезные проблемы у значительного числа людей.

- В элементе BODY можно задать как BGCOLOR, так и BACKGROUND. В этом случае броузер отдает предпочтение BACKGROUND, но если изображение фона невозможно загрузить, будет использовано BGCOLOR. Поэтому старайтесь задавать цвет фона похожим на цвет фонового рисунка, чтобы не нарушился цветовой баланс документа.

2.3. ПРИМЕРЫ

2.3.1. Пример простого HTML-документа

```
<HTML>  
<HEAD>  
<TITLE>Приветствие</TITLE>
```

```

</HEAD>
<BODY>
  <P>Добро пожаловать!</P>
</BODY>
</HTML>

```

Приветствие
Добро пожаловать!

Этот документ отобразится в браузере так:

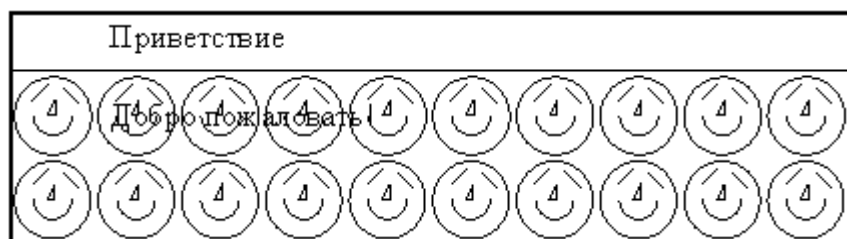
В этом примере обратите внимание на то, куда выводится браузером название документа в элементе TITLE.

2.3.2. Пример использования фонового рисунка

```

<HTML>
<HEAD>
  <TITLE>Приветствие</TITLE>
</HEAD>
<BODY BACKGROUND="smail.gif">
  <P>Добро пожаловать!</P>
</BODY>
</HTML>

```



Этот документ отобразится в браузере так:

3. ЦВЕТОВЫЕ СПЕЦИФИКАЦИИ

Для определения цвета различных элементов HTML-документа необходимо указать значение соответствующих атрибутов. Указывать значения этих атрибутов можно двумя способами:

- определять цвет в формате RGB;
- определять цвет, используя символьную нотацию

3.1. ФОРМАТ RGB

Формат RGB – это система указания цвета, которая базируется на смешении трех основных цветов: красном (RED), зеленом (GREEN) и синем (BLUE). Итоговый цвет определяется цифрами в шестнадцатеричном коде. Для каждого цвета задается шестнадцатеричное значение в пределах от 0 до FF, что соответствует диапазону 0-255 в десятичном исчислении. Затем эти значения объединяются в одно число, перед которым ставится символ #. Например, число #800080 обозначает фиолетовый цвет. Указывая цвет в формате RGB, можно определить более шестнадцати миллионов цветовых оттенков.

3.2. СИМВОЛЬНАЯ НОТАЦИЯ

Задание цвета в формате RGB имеет один недостаток – необходимо помнить совокупности цифр для указания цвета. Этого недостатка лишена символьная нотация. Можно указывать название цвета на английском языке. Но у этого способа указания цвета тоже есть недостаток – определено всего шестнадцать имен цветов.

3.3. СООТВЕТСТВИЕ ФОРМАТА RGB И СИМВОЛЬНОЙ НОТАЦИИ

Ниже приведена таблица соответствий указания цвета в символьной нотации и формате RGB.

Символьная нотация	Формат RGB	Цвет
Black	#000000	Черный
Silver	#C0C0C0	Серебро
Gray	#808080	Серый
White	#FFFFFF	Белый
Maroon	#800000	Темно-бордовый
Red	#FF0000	Красный
Purple	#800080	Фиолетовый
Fuchsia	#FF00FF	Розовый
Green	#008000	Зеленый
Lime	#00FF00	Известь

Olive	#808000	Оливковый
Yellow	#FFFF00	Лимонный
Navy	#000080	Темно-синий
Blue	#0000FF	Синий
Teal	#008080	Темно-бирюзовый
Aqua	#00FFFF	Бирюзовый

Таким образом, строка TEXT="#008000" и строка TEXT="Green" в теге <BODY> одинаково определяют цвет шрифта – зеленый.

4. ВЫВОД ТЕКСТОВОЙ ИНФОРМАЦИИ

Любые тексты, будь то школьные сочинения, заметка в газету или техническое описание устройства, имеют определенную структуру. Элементами такой структуры являются заголовки, подзаголовки, абзацы, списки и др.

Разбиение всего текста на структурные элементы называется логическим форматированием. В HTML-документе логическое форматирование достигается с помощью специальных тегов.

4.1. АБЗАЦЫ

Одним из первых правил составления любых документов является разбиение его текста на отдельные абзацы, выражающие законченную мысль. В HTML-документе разделение на абзацы производится с помощью специального тега <P>. Синтаксис этого тега таков:

```
<P
    ALIGN="выравнивание">
```

Атрибут ALIGN определяет способ выравнивания абзаца. Он может иметь следующие значения:

- LEFT – текст выравнивается по левому краю окна броузера. Это значение используется по умолчанию, т.е. когда атрибут не указан.
- CENTER – текст выравнивается по центру окна броузера.
- RIGHT – текст выравнивается по правому краю окна броузера.

Пример использования тега <P>:

```
<HTML>
<HEAD>
<TITLE>Приветствие</TITLE>
</HEAD>
<BODY>
<P>Добро пожаловать!</P>
<P ALIGN="CENTER">Добро пожаловать!</P>
<P ALIGN="RIGHT">Добро пожаловать!</P>
</BODY>
</HTML>
```


Этот документ отобразится в браузере так:

Приветствие
Добро пожаловать!
Добро пожаловать!
Добро пожаловать!

Браузер автоматически формирует абзацы в зависимости от ширины окна браузера или размера шрифта, перенося слова из строки в строку и отделяя абзацы друг от друга пустой строкой.

4.2. УПРАВЛЕНИЕ ПЕРЕВОДОМ СТРОКИ

Так как браузер автоматически определяет места переноса строк, иногда возникают ситуации запретить перевод строки в каком-нибудь месте или, наоборот, принудительно сделать перевод строки в каком-то определенном месте. Для этого существуют особые теги, управляющие переводом строк.

Когда необходимо сделать принудительный перевод строки, используют тег **
**. Этот тег не имеет атрибутов и закрывающего тега. Пример использования принудительного перевода строки:

```
<HTML>
<HEAD>
  <TITLE>Приветствие</TITLE>
</HEAD>
<BODY>
  <P>Добро<BR>позаловать!</P>
</BODY>
</HTML>
```

Приветствие
Добро позаловать!

Этот пример будет выглядеть так:

При использовании тега **
** пустая строка не образуется, т.е. абзац не прерывается.

В некоторых случаях, наоборот, бывает необходимо сделать так, чтобы браузер не производил перевода строки. Например, не рекомендуется отрывать буквы инициалов от фамилии. В таких случаях тот участок текста, в котором нельзя переводить строку, следует поместить в элемент **NOBR**.

Пример:

```
<P>Это стихотворение написал <NOBR>А.С.
```

```
Пушкин</NOBR> – великий русский поэт.</P>
```

В браузере участок текста “А.С. Пушкин” всегда будет отображаться на одной строке.

Если получится так, что строка, расположенная в элементе NOBR, будет выходить за пределы окна браузера, то внизу окна появится горизонтальная полоса прокрутки.

4.3. ЗАГОЛОВКИ

Почти в каждом тексте используются заголовки для отдельных частей документа. Эти заголовки представляют собой фрагменты текста, которые выделяются на экране при отображении страницы браузером.

Для разметки заголовков используются теги **<H1>**, **<H2>**, **<H3>**, **<H4>**, **<H5>** и **<H6>**. Эти теги требуют соответствующего закрывающего тега. Заголовок с номером 1 является самым крупным (заголовок верхнего уровня), а с номером 6 – самым мелким. Теги заголовка нельзя использовать для выделения отдельных слов текста с целью увеличения их размера. При использовании тегов заголовков происходит вставка пустой строки до и после заголовка, поэтому тегов абзаца и перевода строки здесь не требуется.

Синтаксис тегов заголовков:

<Hn

ALIGN="выравнивание">

Атрибут ALIGN определяет способ выравнивания заголовка. Он может иметь те же значения, что и аналогичный атрибут у тега абзаца.

Пример использования разных заголовков:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>Пример</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<H1>Заголовок 1</H1>
```

```
<H2 ALIGN="CENTER">Заголовок 2</H2>
```

```
<P>Простой текст</P>
```

```
</BODY>
```

```
</HTML>
```

Пример
<p>Заголовок 1</p> <p>Заголовок 2</p> <p>Простой текст</p>

Вот так браузер отобразит данный пример:

4.4. ФОРМАТИРОВАНИЕ ТЕКСТА

В языке HTML предусмотрены специальные теги, предназначенные для форматирования текста. Они позволяют изменять вид шрифта, цвет, раз-мер и др.

Чтобы отобразить текст полужирным шрифтом, используют тег ****. Например:

`<P>Это полужирный шрифт</P>`

Тег **<I>** отображает текст курсивом. Например:

`<P>Выделение <I>курсивом</I></P>`

Используя тег **<TT>**, можно отобразить текст шрифтом, в котором все буквы имеют одинаковую ширину. Это так называемый моноширинный шрифт. Например:

`<P>Это <TT>моноширинный</TT> шрифт</P>`

Для отображения текста подчеркнутым используется тег **<U>**. Например:

`<P>Пример <U>подчеркивания</U> текста</P>`

Тег **<S>** отображает текст, перечеркнутый горизонтальной линией. Например:

`<P>Пример <S>зачеркивания</S> текста</P>`

Тег **<BIG>** выводит текст шрифтом большего (чем непомеченная часть текста) размера. Например:

`<P>Шрифт <BIG>большого</BIG> размера</P>`

Тег **<SMALL>** выводит текст шрифтом меньшего размера. Например:

`<P>Шрифт <SMALL>меньшего</SMALL> размера</P>`

Тег **<SUB>** сдвигает текст ниже уровня строки и выводит его (если возможно) шрифтом меньшего размера. Удобно использовать для математических индексов. Например:

`<P>Шрифт _{нижнего} индекса</P>`

Тег **<SUP>** сдвигает текст выше уровня строки и выводит его (если возможно) шрифтом меньшего размера. Удобно использовать для задания степеней чисел в математике. Например:

`<P>Шрифт ^{верхнего} индекса</P>`

Теги форматирования могут быть вложенными друг в друга. При этом нужно внимательно следить, чтобы один контейнер находился целиком в другом контейнере. Например:

<P>Этот текст <I>полужирный и курсивный</I></P>

Ф ормирование текста
Это полужирный ш риф т
Выделение <i>курсивом</i>
Этo моноширинный ш риф т
Пример <u>подчеркивания</u> текста
Пример зачеркивания текста
Ш риф т бoлшe его размера
Ш риф т мeншe его размера
Ш риф т ^{нижнего} индекса
Ш риф т ^{верхнего} индекса
Этот текст <i>полужирный и курсивный</i>

Вот так браузер отобразит приведенные выше примеры:

4.5.1. Тег

Тег позволяет изменить шрифт для блока текста. Этот тег имеет следующую спецификацию:

```
<FONT
    FACE="тип шрифта"
    COLOR="цвет шрифта"
    SIZE="размер шрифта"
>
```

Атрибут FACE служит для указания типа шрифта, которым браузер будет выводить текст (если такой шрифт имеется на компьютере). Значением данного атрибута служит название шрифта, которое должно в точности совпадать с названием шрифта, имеющимся у пользователя. Если такой шрифт не найдется, то данное указание проигнорируется и будет использован шрифт, установленный по умолчанию.

Можно установить как один, так и несколько названий шрифтов, разделяя их запятыми. Тогда список шрифтов будет просматриваться слева направо так: если на компьютере пользователя нет шрифта, указанного в списке первым, то делается попытка найти следующий и т.д. Если ни одного шрифта найти не удалось, то будет использоваться шрифт, установленный браузером по умолчанию.

Атрибут COLOR устанавливает цвет шрифта. Значение этого атрибута может быть указано в формате RGB или символьной нотацией.

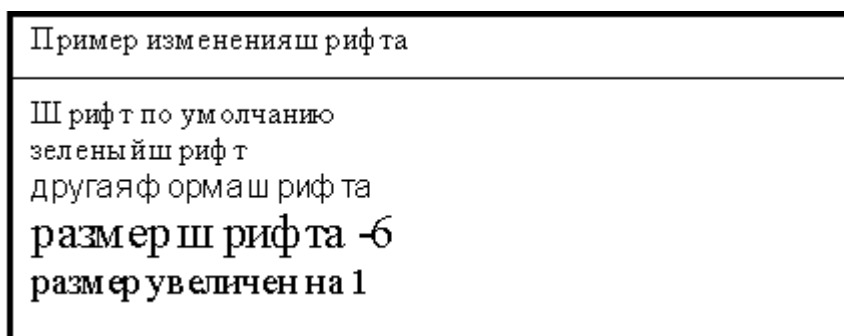
Атрибут SIZE служит для указания размера шрифта. Указывать размер шрифта можно двумя способами: абсолютной величиной или относительной величиной. При указании размера абсолютной величиной значением атрибута

является число от 1 до 7. 1 – самый маленький шрифт, 7 – самый большой. При указании размера относительной величиной значением атрибута является число со знаком + или -. В данном случае шрифт будет увеличен (+) или уменьшен (-) от размера, принятого по умолчанию.

Пример использования тега :

```
<HTML>
<HEAD>
  <TITLE>Пример изменения шрифта</TITLE>
</HEAD>
<BODY>
  <P>Шрифт по умолчанию<BR>
  <FONT COLOR="GREEN">зеленый шрифт</FONT><BR>
  <FONT FACE="Arial">другая форма шрифта</FONT><BR>
  <FONT SIZE="6">размер шрифта – 6</FONT><BR>
  <FONT SIZE="+1">размер увеличен на 1</FONT></P>
</BODY>
</HTML>
```

Этот пример отобразится так:



4.5.2. Контейнер DIV

Иногда бывает необходимо произвести выравнивание большого блока документа, содержащего не только текст, но и рисунки, таблицы и т.п. Для этих целей используется элемент-контейнер **DIV**. Спецификация элемента DIV:

```
<DIV
  ALIGN="выравнивание">
```

Атрибут ALIGN определяет тип выравнивания содержимого и может иметь те же значения, что и элемент P.

4.5.3. Отступы

Иногда требуется отобразить блок текста с отступом. Для этого блок текста помещают в элемент-контейнер **BLOCKQUOTE**. Тогда содержимое этого элемента будет отображаться с небольшими отступами слева и справа, а также отделяться от остального текста пустыми строками.

5. ГИПЕРССЫЛКИ

Как уже было сказано, гиперссылки – переходы к другим документам. Они являются основой HTML. Гиперссылки можно использовать для перехода не только к другим HTML-документам, но и другим объектам, которые можно разместить на компьютере, например, к видео- и аудиофайлам, архивам, рисункам и т.п.

Каждая ссылка состоит из двух частей. Первая – это то, что отображается браузером. Она называется **указателем ссылки**. Вторая часть – адресная, содержащая **адрес** объекта, к которому должен происходить переход. Эту часть называют **URL** (универсальный идентификатор ресурсов).

Когда вы щелкаете мышью по указателю ссылки, браузер загружает документ, адрес которого указан в URL.

Указателем ссылки может быть слово, группа слов или рисунок. Если указатель текстовый, то он обычно отображается браузером подчеркнутым синим шрифтом. При наведении курсора мыши на указатель курсор принимает форму руки, указывая, что это ссылка и можно произвести переход. Если указатель графический, внешне он не отличается от других рисунков, но при наведении курсора мыши на такой указатель, курсор также принимает форму руки.

5.1. УНИВЕРСАЛЬНЫЙ ИДЕНТИФИКАТОР РЕСУРСОВ URL

По своей сути URL – это адрес файла, к которому происходит переход. Указание адреса может быть относительным или абсолютным.

Когда в URL указывается полный путь к файлу, независимо от того, где этот файл находится (в Интернете или на жестком диске компьютера), то это абсолютное указание. Например: **http://www.mysite.ru/page.htm** – абсолютный адрес документа, находящегося в Интернете, **c:\web\doc1.htm** – абсолютный адрес документа на диске **c**.

Если в URL указывается не полный путь, а путь относительно адреса документа, в котором находится ссылка, то это относительное указание. Например, браузер отображает документ, абсолютный адрес которого **c:\web\doc1.htm**, в этом документе имеется ссылка с адресом **pict\ris1.jpg**, то это означает, что на самом деле ссылка будет на документ по адресу **c:\web\pict\ris1.jpg**.

Когда надо сослаться на файл, находящийся на другом компьютере, тогда следует пользоваться абсолютным указанием, а если ссылаться на файлы, находящиеся на том же компьютере, где и документ, содержащий ссылку, то лучше использовать относительное указание.

5.2. ПРАВИЛА ЗАПИСИ ССЫЛОК

Для организации ссылки необходимо указать браузеру, что является указателем ссылки, а также определить адрес документа, на который происходит ссылка. Оба этих действия выполняются с помощью тега `<A>`.

Тег `<A>` имеет следующую спецификацию:

```
<A
  HREF="URL-адрес"
  NAME="имя ссылки"
  TARGET="объект для вывода"
>
```

Атрибут `HREF` используется для задания адреса файла, к которому производится переход. Значением этого атрибута является текстовая строка, содержащая абсолютный или относительный URL-адрес.

Атрибут `NAME` предназначен для задания ссылке имени. Значением этого атрибута является короткая текстовая строка. Этот атрибут используется для ссылок внутри одного HTML-документа.

Атрибут `TARGET` позволяет определить, куда будет выводиться документ, на который происходит переход. Этот атрибут может иметь значение `_blank` – это означает, что документ будет выводиться в новом окне браузера.

Пример ссылки:

```
<A HREF="doc1.htm">Документ 1</A>
```

Браузер отобразит эту строку так:

Документ 1

При нажатии мышью на этой строке браузер загрузит и отобразит файл `doc1.htm`, т.е. “Документ 1” – это указатель ссылки, а “`doc1.htm`” – URL-адрес.

5.3. ВНУТРЕННИЕ ССЫЛКИ

Кроме ссылок на другие документы, часто бывает полезно включить ссылки на различные части текущего документа. Например, большой документ читается лучше, если он имеет оглавление со ссылками на соответствующие разделы.

Для построения внутренней ссылки сначала нужно создать указатель, определяющий место назначения. Для этого в месте, куда потом будет производиться ссылка, надо поместить тег `<A>` с атрибутом `NAME`, определив этим атрибутом имя указателя. Например:

```
<A NAME="glava5"></A>
```

Обратите внимание, что в этом примере отсутствует содержимое тега `<A>`. Обычно так и делают, поскольку здесь нет необходимости как-то выделять текст, а требуется лишь указать местоположение.

После того как место назначения определено, можно приступить к созданию ссылки на него. Для этого в атрибуте `HREF` тега `<A>` помещается имя ссылки с префиксом `#`, говорящим о том, что это внутренняя ссылка. Например:

```
<A HREF="#glava5">Глава 5</A>
```

Теперь, если пользователь щелкнет кнопкой мыши на словах “Глава 5”, то браузер выведет соответствующую часть документа в окне просмотра.

Можно совмещать внутренние ссылки со ссылками на другие документы. Например:

`Глава 5 Документа 1`

При нажатии на эту ссылку браузер откроет файл doc1.htm, найдет в этом файле указатель glava5 и выведет в окне просмотра соответствующую информацию.

5.4. ССЫЛКИ НА ДОКУМЕНТЫ РАЗЛИЧНЫХ ТИПОВ

Когда пользователь щелкает мышью на ссылке, указывающей на другой HTML-документ, то документ выводится непосредственно в окне браузера. Если же ссылка указывает на документ иного типа, программа просмотра принимает документ и затем решает, что с ним делать дальше.

- Браузер знает этот тип документа и умеет с ним обращаться. Например, если создать ссылку на графический файл формата GIF и щелкнуть мышью на ней, то браузер очистит окно просмотра и отобразит указанное изображение.

- Браузер не распознает тип принятого документа и не знает, что с ним делать дальше. В этом случае он обратится к вспомогательным программам, имеющимся на компьютере пользователя. Если подходящая программа найдется, браузер запустит ее и передаст ей полученный документ для обработки. Например, если пользователь щелкнет на ссылке на видеофайл формата AVI, браузер загрузит файл, найдет программу для демонстрации AVI-файлов и запустит ее. Видеофайл будет показан в дополнительном небольшом окне.

5.5. ССЫЛКИ НА РЕСУРСЫ ИНТЕРНЕТА

Основное назначение HTML-документов – это глобальная компьютерная сеть Интернет. HTML-документ, размещенный в Интернете, становится Web-страницей. Чтобы обратиться к Web-странице, надо указать URL-адрес в такой форме: **http://sitename/docname**, где sitename – имя сайта, docname – имя документа. Например: `http://kotoz.newmail.ru/autor.htm`.

Можно на Web-странице разместить адрес электронной почты. Для этого URL-адрес указывается так: **mailto:address**, где address – это адрес почтового ящика. Например: `mailto:vasya@mail.ru`.

Также существуют специальные форматы URL-адреса для других ресурсов Интернета (FTP, TelNet, Newsgroup, Gopher, WAIS).

6. ГРАФИЧЕСКИЕ ЭЛЕМЕНТЫ

Одним из достоинств HTML-документа является возможность использования графических элементов в оформлении. Можно выделить три элемента, чаще всего используемых в HTML-документах: горизонтальные линии, таблицы и рисунки.

6.1. ГОРИЗОНТАЛЬНЫЕ ЛИНИИ

Горизонтальные линии визуально подчеркивают законченность той или иной области документа. Сейчас часто используют рельефную, вдавленную линию, чтобы обозначить “объемность” документа.

Тег **<HR>** позволяет провести рельефную горизонтальную линию в окне большинства браузеров. Этот тег не является контейнером, поэтому не требует закрывающего тега. До и после линии автоматически вставляется пустая строка. Спецификация тега **<HR>**:

```
<HR  
  ALIGN="выравнивание"  
  WIDTH="длина линии"  
  SIZE="толщина линии"  
  NOSHADE  
>
```

Атрибут **ALIGN** определяет способ выравнивания линии. Он может иметь следующие значения:

- **LEFT** – линия выравнивается по левому краю окна браузера. Это значение используется по умолчанию.
- **CENTER** – линия выравнивается по центру окна браузера.
- **RIGHT** – линия выравнивается по правому краю окна браузера.

Атрибут **WIDTH** задает длину линии. Значением данного атрибута является число. Это число означает длину линии в пикселях. Если после числа стоит знак %, то это означает длину в процентах от ширины окна. Например:

<HR WIDTH="400"> – линия длиной 400 пикселей.

<HR WIDTH="50%"> – линия длиной 50 процентов от ширины окна.

Атрибут **SIZE** задает толщину линии. Значением этого атрибута является число. Это число означает толщину линии в пикселях.

Атрибут **NOSHADE** отменяет “трехмерность” линии.

6.2. РИСУНКИ

Без иллюстраций документ скучен, вял и однообразен. HTML позволяет использовать рисунки в формате **JPG** и **GIF** для оформления HTML-документов. Для вставки рисунков используется тег ****. Спецификация тега ****:

```
<IMG  
  SRC="адрес рисунка"  
  ALIGN="выравнивание"  
  HEIGHT="высота рисунка"  
  WIDTH="ширина рисунка"  
  BORDER="толщина рамки"  
>
```

Атрибут SRC определяет URL-адрес рисунка, который будет отображаться браузером.

Атрибут ALIGN определяет способ выравнивания рисунка. Он может иметь следующие значения:

- TOP – рисунок выравнивается по верхнему краю текущей строки.
- MIDDLE – рисунок выравнивается серединой по текущей строке.
- BOTTOM – рисунок выравнивается по нижнему краю текущей строки.
- LEFT – рисунок прижимается к левому краю окна браузера и обтекает текстом.
- RIGHT – рисунок прижимается к правому краю окна браузера и обтекает текстом.

Атрибут HEIGHT определяет высоту рисунка в пикселях.

Атрибут WIDTH определяет ширину рисунка в пикселях.

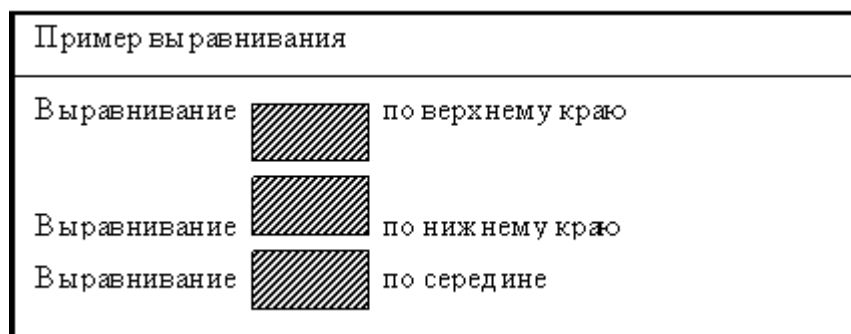
Используя атрибуты HEIGHT и WIDTH можно увеличивать или уменьшать рисунок. Если указать только один из этих атрибутов, то рисунок будет увеличен или уменьшен пропорционально и по ширине, и по высоте.

Атрибут BORDER позволяет задавать рамку вокруг рисунка. Значение этого атрибута – толщина рамки в пикселях. По умолчанию – 1.

Пример выравнивания рисунков:

```
<HTML>
<HEAD>
<TITLE>Пример выравнивания</TITLE>
</HEAD>
<BODY>
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="TOP">по
верхнему краю</P>
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="BOTTOM">
по нижнему краю</P>
<P>Выравнивание<IMG SRC="ris1.jpg" ALIGN="MIDDLE">
по середине</P>
</BODY>
</HTML>
```

Браузер отобразит данный пример так:



6.3 РИСУНОК-ССЫЛКА

Можно использовать рисунки в качестве гиперссылок. Для этого нужно включить тег `IMG` в содержание элемента `A`. Например:

```
<A HREF="glava5.htm"><IMG SRC="ris1.jpg"></A>
```

7 СПИСКИ

В языке HTML предусмотрен специальный набор тегов для представления информации в виде списков. Списки являются одним из наиболее часто употребляемых форм представления данных как в электронных документах, так и печатных. В языке HTML предусмотрены маркированные, нумерованные списки и списки определений.

7.1. МАРКИРОВАННЫЙ СПИСОК

Этот список еще называется нумерованным или неупорядоченным. В маркированном списке для выделения его элементов используются специальные символы, называемые маркерами списка. Вид маркеров определяется браузером, причем при создании вложенных списков браузеры автоматически разнообразят вид маркеров различного уровня вложенности.

Для создания маркированного списка необходимо использовать тег-контейнер `` ``, внутри которого располагаются все элементы списка. Открывающий и закрывающий теги списка обеспечивают перевод строки до и после списка, отделяя, таким образом, список от основного содержимого документа, поэтому нет необходимости использовать теги абзаца или принудительного перевода строки.

Каждый элемент списка должен начинаться тегом `` и заканчиваться тегом ``.

Спецификация элемента `UL`:

```
<UL  
  TYPE="вид маркера">
```

Атрибут `TYPE` задает вид маркера, которым выделяются элементы списка. Он может иметь следующие значения:

- `DISK` – маркеры отображаются закрашенными кружочками, это значение используется по умолчанию;
- `CIRCLE` – маркеры отображаются не закрашенными кружочками;
- `SQUARE` – маркеры отображаются квадратиками.

Спецификация элемента `LI` для маркированного списка:

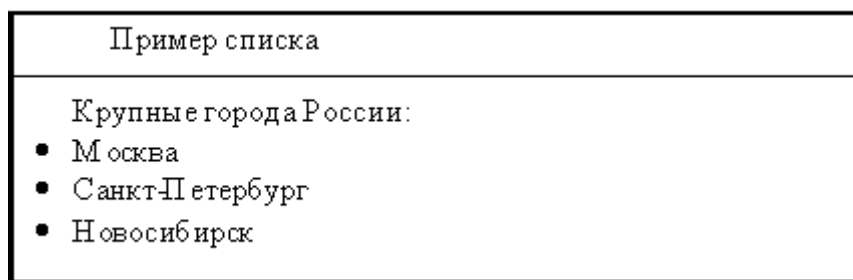
```
<LI  
  TYPE="вид маркера">
```

Атрибут TYPE задает вид маркера, он может принимать такие же значения, что и одноименный атрибут элемента UL. Значение по умолчанию – DISK.

Пример использования маркированного списка:

```
<HTML>
<HEAD>
  <TITLE>Пример списка</TITLE>
</HEAD>
<BODY>
  <UL>
    Крупные города России:
    <LI>Москва</LI>
    <LI>Санкт-Петербург</LI>
    <LI>Новосибирск</LI>
  </UL>
</BODY>
</HTML>
```

Вот так браузер отобразит данный пример:



7.2. НУМЕРОВАННЫЙ СПИСОК

Нумерованные списки иногда называют упорядоченными. Списки данного типа представляют собой упорядоченную последовательность отдельных элементов. Отличием от маркированных списков является то, что в нумерованном списке перед каждым его элементом автоматически проставляется порядковый номер. Вид нумерации зависит от браузера и может задаваться атрибутами тегов списка. В остальном реализация нумерованного списка во многом похожа на реализацию маркированного списка.

Для создания нумерованного списка следует использовать тег-контейнер **** ****, внутри которого располагаются все элементы списка.

Каждый элемент списка должен начинаться тегом **** и заканчиваться тегом ****.

Спецификация элемента OL:

```
<OL
  TYPE="вид нумерации"
  START="начальная позиция">
```

Атрибут TYPE задает вид нумерации, которой выделяются элементы списка. Он может иметь следующие значения:

- A – маркеры в виде прописных латинских букв;
- a – маркеры в виде строчных латинских букв;
- I- маркеры в виде больших римских цифр;
- i- маркеры в виде маленьких римских цифр;
- 1- маркеры в виде арабских цифр, это значение используется по умолчанию.

Атрибут START определяет позицию, с которой начинается нумерация списка. Используя этот атрибут, можно начать нумерацию, например, с цифры 5 или буквы E, в зависимости от вида нумерации. Значением атрибута START является число, вне зависимости от вида нумерации.

Спецификация элемента LI для нумерованного списка:

```
<LI  
  TYPE="вид нумерации"  
  VALUE="номер элемента">
```

Атрибут TYPE задает вид нумерации, он может принимать такие же значения, что и одноименный атрибут элемента OL. По умолчанию значение этого атрибута – 1.

Атрибут VALUE позволяет изменить номер данного элемента, при этом изменятся номера и последующих элементов.

Пример использования нумерованного списка:

```
<HTML>  
<HEAD>  
  <TITLE>Пример списка</TITLE>  
</HEAD>  
<BODY>  
  <UL>  
    Города России по величине:  
    <LI>Москва</LI>  
    <LI>Санкт-Петербург</LI>  
    <LI>Новосибирск</LI>  
  </UL>  
</BODY>  
</HTML>
```

Вот так браузер отобразит данный пример:

Пример списка
Города России по величине: 1. Москва 2. Санкт-Петербург 3. Новосибирск

7.3. СПИСОК ОПРЕДЕЛЕНИЙ

Списки определений, также называемые словарями определений специальных терминов, являются особым видом списков. В отличие от других типов списков каждый элемент списка определений всегда состоит из двух частей. В первой части элемента списка указывается определяемый термин, а во второй части – текст в форме словарной статьи, раскрывающий значение термина.

Списки определений задаются с помощью тега-контейнера **<DL>**. Внутри него тегом **<DT>** отмечается определяемый термин, а тегом **<DD>** – абзац с его определением. Внутри элемента **<DT>** нельзя использовать абзацы (P) и заголовки (H1-H6), но их можно использовать внутри элемента **<DD>**. Атрибутов для элементов списка определений нет. В общем случае список определений записывается следующим образом:

```
<DL>
<DT>Термин</DT>
<DD>Определение термина</DD>
</DL>
```

Пример использования списка определений:

```
<HTML>
<HEAD>
<TITLE>Пример списка определений</TITLE>
</HEAD>
<BODY>
<DL>
<H2 ALIGN="CENTER">Состав Microsoft Office</H2>
<DT>Microsoft Word</DT>
<DD>Многофункциональный текстовый процессор</DD>
<DT>Microsoft Excel</DT>
<DD>Программа для работы с электронными
таблицами</DD>
<DT>Microsoft Access</DT>
<DD>Система управления базами данных</DD>
</DL>
</BODY>
</HTML>
```

Этот пример отобразится в браузере так:

Пример списка определений
<p style="text-align: center;">Состав Microsoft Office</p> <p>Microsoft Word Многофункциональный текстовый процессор</p> <p>Microsoft Excel Программа для работы с электронными таблицами</p> <p>Microsoft Access Система управления базами данных</p>

8 ТАБЛИЦЫ

Одним из наиболее мощных и широко применяемых в HTML средств являются таблицы. Они используются не только традиционно как метод представления данных, но и как средство форматирования Web-страниц. Документ HTML может содержать произвольное число таблиц, причем допускается вложенность таблиц друг в друга.

Каждая таблица начинается тегом **<TABLE>** и заканчивается тегом **</TABLE>**. Внутри этой пары тегов располагается описание содержимого таблицы. Любая таблица состоит из одной или нескольких строк, в которых задаются данные для отдельных ячеек.

Каждая строка начинается тегом **<TR>** и заканчивается тегом **</TR>**. Отдельная ячейка в строке обрамляется парой тегов **<TD>** и **</TD>** или **<TH>** и **</TH>**. Тег **<TH>** используется для ячеек заголовка таблицы, а **<TD>** – для ячеек данных. Отличие этих тегов в том, что в заголовке по умолчанию используется полужирный шрифт, а для данных – обычный.

Теги **<TD>** и **<TH>** не могут появляться вне описания строки таблицы **<TR>**.

Пример таблицы:

```

<HTML>
<HEAD>
<TITLE>Пример таблицы</TITLE>
</HEAD>
<BODY>
<TABLE>
<TR>
<TD>Ячейка 1</TD>
<TD>Ячейка 2</TD>
</TR>
<TR>
<TD>Ячейка 3</TD>

```

```

<TD>Ячейка 4</TD>
</TR>
</TABLE>
</BODY>
</HTML>

```

Этот пример отобразится в браузере так:

Пример таблицы	
Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

Спецификация тега <TABLE>:

```

<TABLE
  ALIGN="выравнивание"
  BORDER="толщина рамки"
  CELLPADDING="расстояние"
  CELLSPACING="расстояние"
  HEIGHT="высота"
  VALIGN="вертикальное выравнивание"
  WIDTH="ширина"
>

```

Атрибут ALIGN определяет выравнивание таблицы в окне просмотра браузера. Он может иметь одно из двух значений: LEFT (по левому краю) и RIGHT (по правому краю). По умолчанию – LEFT.

Атрибут BORDER управляет толщиной рамки. Значением этого атрибута является число. Это число определяет толщину рамки таблицы в пикселях. По умолчанию толщина рамки – 1.

Атрибут CELLPADDING определяет расстояние в пикселях между рамкой и содержимым ячейки. По умолчанию – 1.

Атрибут CELLSPACING определяет расстояние в пикселях между ячейками таблицы. По умолчанию – 2.

Атрибут HEIGHT определяет высоту таблицы в пикселях.

Атрибут VALIGN определяет вертикальное выравнивание содержимого таблицы. Он может иметь следующие значения: TOP (по верхнему краю), MIDDLE (посередине) и BOTTOM (по нижнему краю). По умолчанию – MIDDLE.

Атрибут WIDTH определяет ширину таблицы в пикселях или процентах от ширины окна браузера.

Спецификация тега <TR>

```

<TR
  ALIGN="выравнивание"
  BGCOLOR="цвет фона"

```


VALIGN="вертикальное выравнивание"

>

Атрибут ALIGN определяет выравнивание содержимого всех ячеек строки. Он может иметь одно из трех значений: LEFT (по левому краю), RIGHT (по правому краю) и CENTER (по центру). По умолчанию – CENTER.

Атрибут BGCOLOR определяет цвет фона для всех ячеек строки. Его значение можно указывать в символьной нотации или в формате RGB.

Атрибут VALIGN определяет вертикальное выравнивание содержимого всех ячеек строки. Он может иметь следующие значения: TOP (по верхнему краю), MIDDLE (посередине) и BOTTOM (по нижнему краю). По умолчанию – MIDDLE.

Спецификация тега <TD>

<TD

ALIGN="выравнивание"

BGCOLOR="цвет фона"

COLSPAN="количество ячеек"

HEIGHT="высота ячейки"

ROWSPAN=" количество ячеек "

VALIGN="вертикальное выравнивание"

WIDTH="ширина ячейки"

>

Атрибут ALIGN определяет выравнивание содержимого ячейки. Он может иметь одно из трех значений: LEFT (по левому краю), RIGHT (по правому краю) и CENTER (по центру). По умолчанию – CENTER.

Атрибут BGCOLOR определяет цвет фона для ячейки. Его значение можно указывать в символьной нотации или в формате RGB.

Атрибут COLSPAN позволяет объединить несколько соседних ячеек по горизонтали. Значение этого атрибута – количество объединяемых ячеек.

Атрибут HEIGHT определяет высоту ячейки в пикселях.

Атрибут ROWSPAN позволяет объединить несколько соседних ячеек по вертикали. Значение этого атрибута – количество объединяемых ячеек.

Атрибут VALIGN определяет вертикальное выравнивание содержимого ячейки. Он может иметь следующие значения: TOP (по верхнему краю), MIDDLE (посередине) и BOTTOM (по нижнему краю). По умолчанию – MIDDLE.

Атрибут WIDTH определяет ширину ячейки в пикселях.

9 СЛОИ

Слой представляет собой прямоугольный контейнер, в котором могут содержаться текст, рисунки, формы, подключаемые модули (plugins) и даже другие слои, т.е. любые элементы, которые можно поместить в тело HTML-документа. Слои обеспечивают разработчикам web-страниц полный контроль (вплоть до одного пикселя) над размещением элементов на странице.

Единственное, что нельзя сделать, используя слои, - это заставить слои отображаться одинаково в разных браузерах. Дело в том, что в Internet Explorer для описания слоя используется тег-контейнер <div> (иногда), а фирма Netscape для тех же целей изобрела свои собственные теги, не входящие в официальную спецификацию HTML, - <layer> и <ilayer>.

10. КАСКАДНЫЕ (МНОГОУРОВНЕВЫЕ) ТАБЛИЦЫ СТИЛЕЙ

Каскадные (многоуровневые) таблицы стилей это набор правил форматирования тегов HTML. Они дают возможность хранить содержимое отдельно от его представления.

Стиль включает все типы элементов дизайна: шрифт, фон, текст, цвета ссылок, поля и расположение объектов на странице.

CSS предполагает 3 типа таблиц стилей - встроенные, внедренные (внутренние) и связанные (внешние).

Существует три метода для применения таблицы стилей к документу HTML:

- Встроенный (Inline). Этот метод позволяет взять любой тег HTML и добавить к нему стиль. Использование встроенного метода предоставляет максимальный контроль над всеми свойствами Web-страницы. Предположим, вы хотите задать внешний вид отдельного абзаца. Вы можете просто добавить атрибут style к тегу абзаца, и браузер отобразит этот абзац с помощью параметров стиля, добавленного в код.

- Внедренный (Embedded). Внедрение позволяет контролировать всю страницу HTML. При использовании тега <style>, помещенного внутри раздела <head> страницы HTML, в код вставляются детализированные атрибуты стиля, которые будут применяться ко всей странице.

- Связанный (Linked или External). Связанная таблица стилей - мощный инструмент, который позволяет создавать образцы стилей (master styles), которые можно затем применять ко всему узлу. Основным документом таблицы стилей (расширение .css) создается Web-дизайнером. Этот документ содержит стили, которые будут едиными для всего Web-узла (неважно, содержит одну страницу или тысячи страниц). Любая страница, связанная с этим документом, будет использовать указанные стили.

Определение правил CSS

Правила определения CSS состоят из селекторов и описаний их свойств. В качестве селекторов используются стандартные имена тегов, свойства которых необходимо дополнить или изменить. Все, что находится в фигурных скобках принято называть определением (или описанием):

селектор {свойство 1: значение 1; свойство 2: значение 2;...}

Например, для того, чтобы цвет фона Web-страницы сделать черным, необходимо объявить следующее правило форматирования:

body{background:black}

В данном случае объявлено правило форматирования тега body, а именно - свойству стиля background присвоено значение black (черный). В результате применения этого правила цвет фона всего документа изменится на черный.

Изменим с помощью CSS не только цвет фона Web-страницы, но и цвет шрифта (на белый).

```
body{background:black;color:white}
```

Формат самого правила не имеет значения, главное - правило должно читаться удобно и легко. Например, вышеприведенное правило можно записать и так:

```
body{  
background:black;  
color:white}
```

Одно и то же правило стиля можно применить сразу к нескольким различным тегам HTML-страницы. Например:

```
body,td,h1{  
background:black;  
color:white}
```

Порой требуется одному и тому же тегу (элементу) придать разные свойства. Например, в одном случае заголовок типа h2 должен быть выделен зеленым цветом, а в другом этот же тип заголовка - красным. Подобные задачи решаются путем использования классов и идентификаторов. Во внешнем файле следует поместить описание двух классов:

```
h2.green{  
color : green  
}  
h2.red{  
color : red  
}
```

Вызов созданных классов в местах форматирования текста осуществляется так:

```
<h2 class="green">Заголовок будет зеленым</h2>  
<h2 class="red">Заголовок будет красным</h2>
```

Встроенный стиль

Встроенный стиль применяется к любому тегу HTML с помощью атрибута style следующим образом:

```
<P style="font: 12pt Courier New"> The text in this line will  
display as 12 point text using the Courier New font.  
</P>
```

Или:

```
<p style="font: 12pt Arial">  
The text in this line will display as 18 point text using the  
Arial font.
```

</p>

Можно добавлять встроенный стиль в любой тег HTML, в котором эта операция будет иметь смысл. Среди таких тегов можно назвать абзацы, заголовки, горизонтальные линии, якоря и ячейки таблиц. Ко всем этим элементам логично применять встроенные стили.

Существуют два тега, которые помогают применять встроенные стили к разделам страницы. Это теги <div> (division - раздел) и (промежуток).

Эти теги определяют диапазон текста, так что все, находящееся между ними, будет оформлено с помощью нужного стиля. Единственное различие между <div> и состоит в том, что <div> создает принудительный разрыв строки, а - нет.

Следовательно, нужно использовать для изменения стиля любой части текста, меньшей абзаца.

Вот пример работы тега <div>:

```
<div style="font-family: Garamond; font-size: 18 pt;">All of the  
text within this section is 18 point Garamond.
```

All of the text within this section is 18 point Garamond.

и тега :

```
<span style="color:#ff3300;"> This text appears in the color red,  
with no line break after the closing span tag </span> and the rest of  
the text.
```

Внедренный стиль

Внедренные стили используют тег <style>, расположенный между тегами </head> и <body> в стандартном документе HTML.

Рассмотрим пример внедренного стиля:

```
<html>  
<head>  
<title>Embedded Style Sheet Example </title>  
</head>  
<style>  
BODY{  
background: #FFFFFF;  
color: #000000;  
}  
H1{  
font: 14pt verdana; color: #CCCCCC;}  
P{font: 12pt times;}  
A{color: #FF0000; text-decoration: none}  
</style>
```

Связанные таблицы стилей

Связанные (linked), или внешние (external), таблицы стилей являются расширением понятия внедренных стилей. Используется тот же самый код, что и

для внедренной таблицы стилей, но он помещается в отдельный документ (файл с расширением .css). После этого с этим документом (файлом) связываются все страницы, к которым необходимо применить данный стиль.

Вот как выглядит пример связанной таблицы стилей:

```
<style>
BODY{
    background: #ffffcc;
    color: #000000;
}
P {
    font-family : sans-serif;
    font-style : italic;
    font-size : 16pt;
    color: #006633;
}
H1{
    font-family: Arial, Helvetica, sans-serif;
    font-size: 24pt;
    color: #996633;
}
</style>
```

Теперь сохраним этот документ как отдельный файл. Назовем его style-1.css и поместим в папку таблиц стилей с именем style.

С этим документом любое количество страниц HTML. Для этого нужно использовать между тегами </title> и </head> следующую конструкцию:

```
<link rel=stylesheet href="style-1.css" type="text/css">
```

Любая страница, содержащая такую связь, будет оформлена в соответствии со стилями, указанными в файле style_1.css.

Код этой страницы выглядит следующим образом:

```
<html>
<head>
<title>Пример использования связанной таблицы стилей</title>
<link href="style/style-1.css" rel=stylesheet type="text/css">
</head>
```

```
<body>
<h1>Старинная студенческая песня</h1>
<P>
```

```
Поднявши меч на наш союз <br>
достоин будет худшей кары,<br>
и я за жизнь его тогда<br>
не дам и ломанной гитары.<br>
Как вожаденно жаждет век<br>
нащупать брешь у нас в цепочке...<br>
```

```
Возьмемся за руки друзья,<br>
чтоб не пропасть поодиночке.<br>
</P>
```

```
<a href="css.htm#метка_51">Вернемся? </a>
```

```
</body>
</html>
```

Здесь важно помнить о смысле концепции каскадирования. Если вам нужны 10 страниц HTML, на которые глобально воздействует эта таблица стилей, вы можете ее применить. Затем, если нужно внести небольшие корректировки в отдельные страницы, можно либо внедрить в эти страницы дополнительные стили, либо использовать встроенный стиль.

Задания к лабораторной работе

Разработать три взаимосвязанных HTML-страницы для размещения на сайте. Первый HTML-документ предназначен для размещения на сайте в качестве первой страницы (имеющей обычно имя index.html, default.html или home.html).

На странице должны располагаться следующие элементы:

- название фирмы;
- логотип фирмы;
- обращение к посетителю страницы;
- адрес фирмы;
- HTML-ссылки на две другие страницы сайта.

Второй HTML-документ предназначен для размещения простейшей рекламы товаров. На странице должны располагаться следующие элементы:

- название фирмы;
- логотип фирмы;
- обращение к посетителю страницы;
- прайс-лист в форме таблицы;
- HTML-ссылки на две другие страницы сайта.

Третий HTML-документ предназначен для размещения перечня услуг/товаров фирмы. На странице должны располагаться следующие элементы:

- заголовок “Перечень услуг/товаров” с указанием имени или фамилии учащегося;
- логотип фирмы;
- перечень;
- HTML-ссылки на две другие страницы сайта.

При разработке HTML-страницы использовать указанную CSS-таблицу.

Вариант1

Название фирмы - заголовок пятого уровня, выравнивается по центру края, буквы желтого цвета.

Остальной текст выравнивается по ширине, шрифт фиолетового цвета, начертание - полужирный, размер шрифта на 2 единицы больше текущего.

Таблица выровнена по правому краю, ширины рамки 3 пикселя, расстояние между границами ячеек равно 3, цвет заголовков - аквамарин.

Список нумерованный, нумерация арабскими цифрами.

Для задания свойств списка использовать внедренную таблицу стилей

Вариант2

Название фирмы - заголовок первого уровня, выравнивается по центру, буквы красного цвета.

Остальной текст по правому краю, шрифт синего цвета, начертание - курсив, размер шрифта на 2 единицы больше текущего.

Таблица выровнена по правому краю, ширины рамки 3 пикселя, расстояние между границами ячеек равно 4.

Список маркированный, маркеры – незакрашенные кружки.

Для задания свойств таблицы использовать внедренную таблицу стилей

Вариант3

Название фирмы - заголовок первого уровня, выравнивается по правому краю, буквы красного цвета.

Остальной текст выравнивается по правому краю, шрифт зеленого цвета, начертание - подчеркнутый, размер шрифта максимально возможный.

Таблица расположена по центру, с заголовком, расположенным над таблицей, ширины рамки 2 пикселя, ячейки, в которых расположен заголовок, голубого цвета.

Список многоуровневый. Тип маркеров каждого уровня выбрать самостоятельно.

Для задания свойств шрифта основного текста использовать внешнюю таблицу стилей.

Вариант4

Название фирмы - заголовок третьего уровня, выравнивается по правому краю, буквы желтого цвета.

Остальной текст выравнивается по центру, шрифт зеленого цвета, начертание - курсив, размер шрифта на 1 единицу больше текущего.

Таблица расположена по центру, с заголовком, расположенным над таблицей, ширины рамки 2 пикселя, ячейки, в которых расположен заголовок, желтого цвета.

Список маркированный, маркеры – закрашенные квадратики.

Для задания свойств списка использовать внешнюю таблицу стилей

Вариант5

Название фирмы - заголовок первого уровня, выравнивается по левому краю, буквы зеленого цвета.

Остальной текст по правому краю, шрифт синего цвета, начертание - курсив, размер шрифта на 2 единицы больше текущего. Таблица расположена по центру, с заголовком, расположенным над таблицей, ширины рамки 2 пикселя, ячейки, в которых расположен заголовок, серого цвета.

Список маркированный, маркеры – закрашенные кружки.

Для задания свойств таблицы использовать внешнюю таблицу стилей.