

HTML

HTML	1
Задание 1. Создание простейшего html документа	1
Задание 2. Форматирование шрифта html документа	7
Задание 3. Списки и ссылки	14
Задание 4. Вставка объектов и работа с цветовым оформлением документа	19
Задание 5. Создание таблиц, вставка звука и видео	23
Задание 6. Контрольная работа по созданию html документа	28
Приложение: Список тем для контента страниц по вариантам	29
Темы по информационным системам и технологиям	29
Темы по праву и организации социального обеспечения	30

Задание 1. Создание простейшего html документа

Чтобы опубликовать документ в глобальной сети Интернет, достаточно поместить его на сервер, постоянно подключенный к сети Интернет и способный общаться с другими серверами с помощью протокола передачи гипертекстов (HyperText Transfer Protocol, или http://). Совокупность таких серверов получила название "всемирной паутины" (World Wide Web, или WWW).

Разработку документов, однако, можно проводить и на компьютере, не имеющем выхода в Интернет.

Что такое HTML?

Термин HTML (HyperText Markup Language) означает "язык маркировки гипертекстов". Первую версию HTML разработал сотрудник Европейской лаборатории физики элементарных частиц Тим Бернерс-Ли.

Для создания HTML документа нам потребуется:

Любой браузер, т.е., программа, пригодная для просмотра HTML-файлов, например Internet Explorer.

И любой редактор текстовых файлов, например Блокнот.

Мы будем использовать текстовый редактор для подготовки HTML-файлов, а браузер - как инструмент контроля за сделанным.

Один и тот же *.html-файл может быть одновременно открыт и в Блокноте, и в Internet Explorer. Сохранив изменения в Блокноте, просто нажмите кнопку «Обновить» в Internet Explorer, чтобы увидеть эти изменения реализованными в HTML-документе.

После того, как вы приобретёте первичные навыки создания страниц, вы сможете подобрать себе редактор HTML-документов по вкусу: Adobe Dreamweaver, Microsoft FrontPage, AkelPad, Macromedia HomeSite, Bluefish, Notepad++, Microsoft Office SharePoint Designer либо иной.

Как устроен HTML-документ

HTML-документ — это просто текстовый файл с расширением *.html, *.htm (Unix-системы могут содержать файлы с расширением *.html). Создадим простой HTML-документ.

Запустите Блокнот и введите в новый документ текст, представленный ниже между пунктирными линиями

```
-----  
<html>  
<head>  
<title>  
Пример 1 Андрей Петров, 34 группа  
<!-- Естественно, Вы должны вставить сюда свои собственные данные -->  
</title>  
</head>  
<body>  
<H1>  
Привет!  
</H1>  
<P>  
Это простейший пример HTML-документа.  
</P>  
<P>  
Этот *.html-файл может быть
```

*одновременно открыт и в Блокноте, и в Internet Explorer.
Сохранив изменения в Блокноте, просто нажмите
кнопку «Обновить» в Internet Explorer или F5,
чтобы увидеть эти изменения реализованными*

в HTML-документе.

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

Примечание. Отступы в тексте сделаны для удобства чтения, при разработке документа их делать не обязательно.

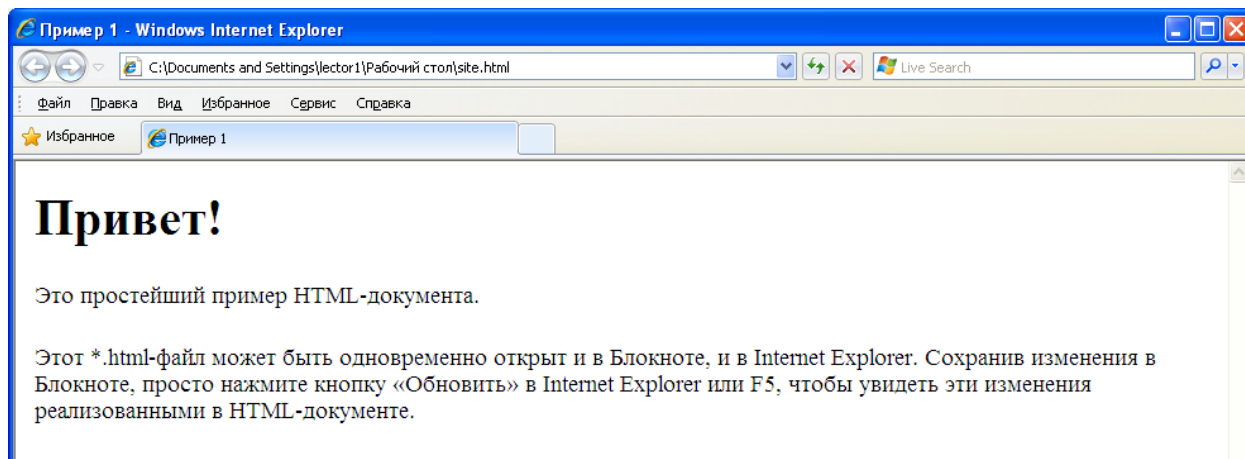
Сохраните этот файл с именем Пример1.html в папке со своим именем. Для этого выберите команду сохранить как... В открывшемся окне папок выберите папку со своим именем. В поле имени файла введите Пример1.html, а в поле тип файла выберите «все файлы».

Ок.

Теперь посмотрим что получилось.

Откройте папку со своим именем и дважды щелкните по файлу Пример1.html.

Если в результате вы видите окно подобное представленному на рисунке, значит вы все сделали правильно.



Как видно из примера, вся информация о форматировании документа сосредоточена в его фрагментах, заключенных между знаками "<" и ">". Такой фрагмент (например, <html>) называется меткой (по-английски — tag, читается "тэг").

Большинство HTML-меток — парные, то есть на каждую открывающую метку вида `<tag>` есть закрывающая метка вида `</tag>` с тем же именем, но с добавлением `"/`.

Метки можно вводить как большими, так и маленькими буквами. Например, метки `<body>`, `<BODY>` и `<Body>` будут восприняты браузером одинаково.

Многие метки, помимо имени, могут содержать атрибуты — элементы, дающие дополнительную информацию о том, как браузер должен обработать текущую метку. В нашем простейшем документе, однако, нет ни одного атрибута. Но мы обязательно встретимся с атрибутами позже.

Метки `<html>` ... `</html>`

Метка `<html>` должна открывать HTML-документ. Аналогично, метка `</html>` должна завершать HTML-документ.

`<head>` ... `</head>`

Эта пара меток указывает на начало и конец заголовка документа. Помимо наименования документа (см. описание метки `<title>` ниже), в этот раздел может включаться множество служебной информации, о которой мы обязательно поговорим чуть позже.

`<title>` ... `</title>`

Все, что находится между метками `<title>` и `</title>`, толкуется браузером как название документа. Internet Explorer, например, показывает название текущего документа в названии вкладки и печатает его в левом верхнем углу каждой страницы при выводе на принтер. Рекомендуется название не длиннее 64 символов.

`<body>` ... `</body>`

Эта пара меток указывает на начало и конец тела HTML-документа, каковое тело, собственно, и определяет содержание документа.

`<H1>` ... `</H1>` — `<H6>` ... `</H6>`

Метки вида <H_i> (где *i* — цифра от 1 до 6) описывают заголовки шести различных уровней. Заголовок первого уровня — самый крупный, шестого уровня, естественно — самый мелкий.

<P> ... </P>

Такая пара меток описывает абзац. Все, что заключено между <P> и </P>, воспринимается как один абзац.

Метки <H_i> и <P> могут содержать дополнительный атрибут ALIGN (читается "элайн", от английского "выравнивать"), например:

<H1 ALIGN=CENTER>Выравнивание заголовка по центру</H1>

или

<P ALIGN=RIGHT>Образец абзаца с выравниванием по правому краю</P>

Дополним наш пример новыми элементами.

Откройте в **Блокноте** Пример1. (Браузер закрывать не надо)

Добавим перед закрывающей меткой </body> такой текст:

<H1 ALIGN=CENTER>Ещё раз привет!</H1>

<H2>Это чуть более сложный пример HTML-документа</H2>

<P>Теперь мы знаем, что абзац можно выравнивать не только влево, </P>

<P ALIGN=CENTER>но и по центру</P>

<P ALIGN=RIGHT>или по правому краю.</P>

Выполните команду «Сохранить» из меню Файл в Блокноте, перейдите в окно браузера и нажмите в нем кнопку обновить (или F5).

Посмотрите на получившийся результат.

Примечание. Если в окне браузера у вас видны теги, значит вы допустили какую-то ошибку. Проверьте введённый текст ещё раз.

С этого момента Вы знаете достаточно, чтобы создавать простые HTML-документы самостоятельно от начала до конца.

Попробуем улучшить наш документ. Начнем с малого — с абзаца.

Непарные метки

Рассмотрим метки, которые не имеют пары, а некоторые (так называемые &-последовательности) к тому же должны вводиться **ТОЛЬКО маленькими** буквами.

Эта метка используется, если необходимо перейти на новую строку, не прерывая абзаца. Очень удобно при публикации стихов.

Добавьте к нашему документу перед меткой </body> такой текст:

*<H1>Стих</H1>
<H2>Автор неизвестен</H2>
<P> Я вчера поймал козла

Привязал на три узла: .

Два морских, и бантик -

Я вообще романтик! </P>*

<HR>

Метка **<HR>** описывает горизонтальную линию

Метка может дополнительно включать атрибуты **SIZE** (определяет толщину линии в пикселах) и/или **WIDTH** (определяет размах линии в процентах от ширины экрана).

Добавьте к нашему документу перед меткой **</body>** небольшую коллекцию горизонтальных линий:

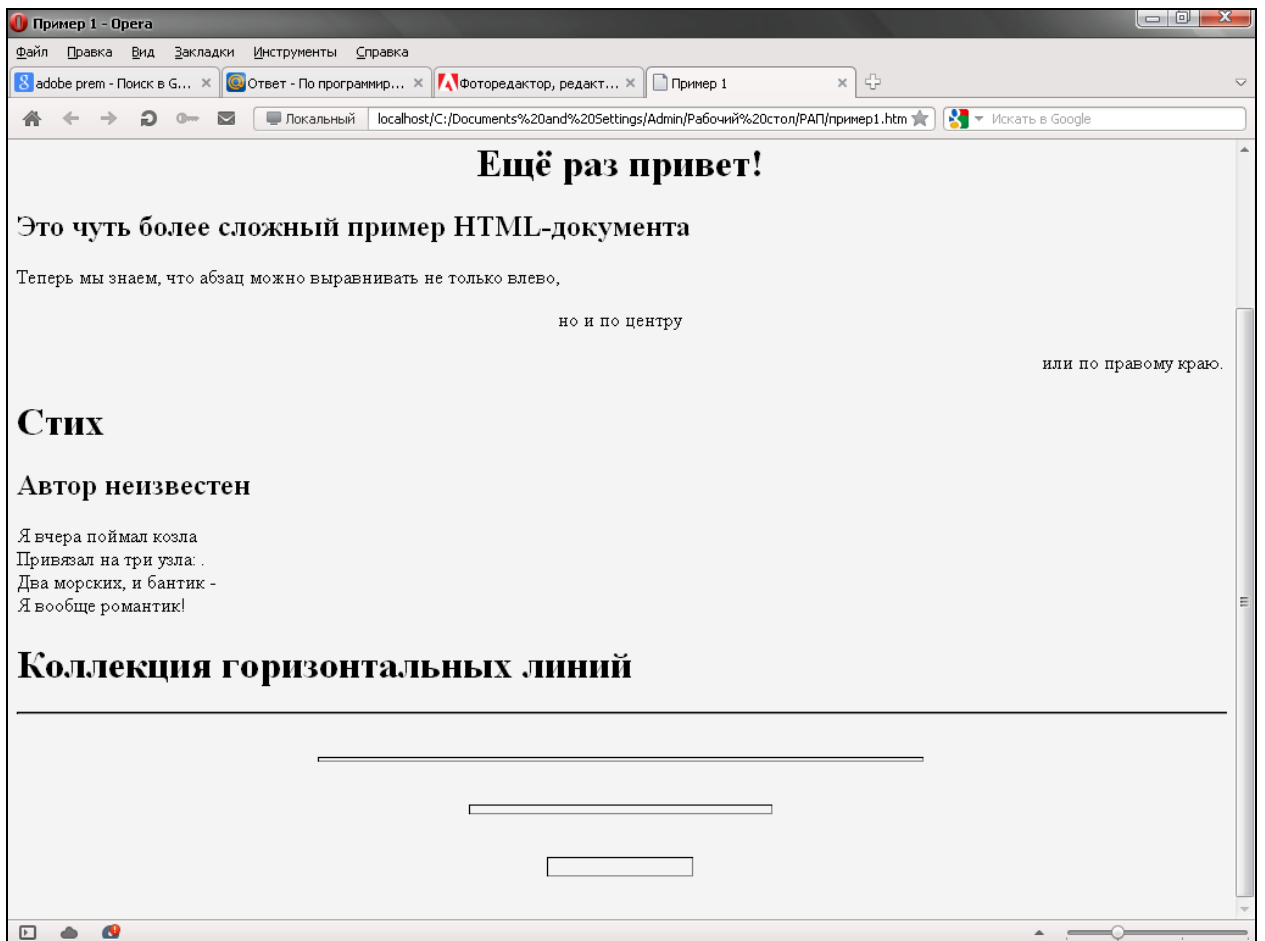
*<H1>Коллекция горизонтальных линий</H1>
<HR SIZE=2 WIDTH=100%>

<HR SIZE=4 WIDTH=50%>

<HR SIZE=8 WIDTH=25%>

<HR SIZE=16 WIDTH=12%>
*

и покажите получившийся результат преподавателю.



Задание 2. Форматирование шрифта html документа

&-последовательности

Поскольку символы "<" и ">" воспринимаются браузерами как начало и конец HTML-меток, возникает вопрос: а как показать эти символы на экране? В HTML это делается с помощью &-последовательностей (их еще называют символьными объектами или эскейп-последовательностями). Браузер показывает на экране символ "<", когда встречает в тексте последовательность < (по первым буквам английских слов less than — меньше, чем). Знак ">" кодируется последовательностью > (по первым буквам английских слов greater than — больше, чем).

Символ "&" (амперсанд) кодируется последовательностью &

Двойные кавычки (") кодируются последовательностью "

Примечание: точка с запятой — обязательный элемент &-последовательности. Кроме того, все буквы, составляющие последовательность, должны быть в нижнем регистре (т.е., маленькие). Использование меток типа " или & не допускается.

Вообще говоря, &-последовательности определены для всех символов из второй половины ASCII-таблицы (куда, естественно, входят и русские буквы). Но использовать мы из будем только для вывода специальных символов.

Комментарии

Браузеры игнорируют любой текст, помещенный между <!-- и -->. Это удобно для размещения комментариев:

<!-- Это комментарий -->

Форматирование шрифта

HTML допускает два подхода к шрифтовому выделению фрагментов текста. С одной стороны, можно прямо указать, что шрифт на некотором участке текста должен быть жирным или наклонным, то есть изменить физический стиль текста. С другой стороны, можно пометить некоторый фрагмент текста как имеющий некоторый отличный от нормального логический стиль, оставив интерпретацию этого стиля браузеру. Поясним это на примерах.

Физические стили

Под физическом стилем принято понимать прямое указание браузеру на модификацию текущего шрифта. Например, все, что находится между метками и , будет написано жирным шрифтом. Текст между метками <I> и </I> будет написан наклонным шрифтом.

Несколько особняком стоит пара меток <TT> и </TT>. Текст, размещенный между этими метками, будет написан шрифтом, имитирующим пишущую машинку, то есть имеющим фиксированную ширину символа.

Логические стили

При использовании логических стилей автор документа не может знать заранее, что увидит на экране читатель. Разные браузеры толкуют одни и те

же метки логических стилей по-разному. Некоторые браузеры игнорируют некоторые метки вообще и показывают нормальный текст вместо выделенного логическим стилем. Вот самые распространенные логические стили.

** ... **

От английского emphasis — акцент.

** ... **

От английского strong emphasis — сильный акцент.

<CODE> ... </CODE>

Рекомендуется использовать для фрагментов исходных текстов.

<SAMP> ... </SAMP>

От английского sample — образец. Рекомендуется использовать для демонстрации образцов сообщений, выводимых на экран программами.

<KBD> ... </KBD>

От английского keyboard — клавиатура. Рекомендуется использовать для указания того, что нужно ввести с клавиатуры.

<VAR> ... </VAR>

От английского variable — переменная. Рекомендуется использовать для написания имен переменных.

Подытожим наши знания о логических и физических стилях с помощью примера. Заодно Вы сможете увидеть, как Ваш браузер показывает те или иные логические стили.

Создадим следующий документ:

```
-----  
<html>  
<head>  
<title>  
Стили. Андрей Петров, 34 группа  
<!-- Естественно, Вы должны вставить сюда свои собственные данные -->  
</title>  
</head>  
<body>  
<H1>  
Шрифтовое выделение фрагментов текста  
</H1>  
<P>  
Теперь мы знаем, что фрагменты текста можно  
выделять  
<B>жирным</B> или <I>наклонным</I> шрифтом.  
Кроме того, можно
```

включать в текст фрагменты с фиксированной шириной символа

```
<TT>(имитация пишущей машинки)</TT></P>
<P>Кроме того, существует ряд логических стилей:</P>
<P><EM>EM - от английского emphasis - акцент </EM><BR>
<STRONG>STRONG - от английского strong emphasis -
сильный акцент </STRONG><BR>
<CODE>CODE - для фрагментов
исходных текстов</CODE><BR>
<SAMP>SAMP - от английского sample -
образец </SAMP><BR>
<KBD>KBD - от английского keyboard -
клавиатура</KBD><BR>
<VAR>VAR - от английского variable -
переменная </VAR></P>
</body>
</html>
```

HTML позволяет определять внешний вид целых абзацев текста. Абзацы можно организовывать в списки, выводить их на экран в отформатированном виде, или увеличивать левое поле. Разберем все по порядку.

**Ненумерованные списки: ... **

Текст, расположенный между метками и , воспринимается как ненумерованный список. Каждый новый элемент списка следует начинать с метки . Например, чтобы создать вот такой список:

- клавиатура;
- монитор;
- мышь.

необходимо перед меткой </body> ввести вот такой HTML-текст:

```
<H2>СПИСКИ</H2>
<UL>
<LI> клавиатура;
<LI> монитор;
<LI> мышь.
</UL>
```

Обратите внимание: у метки нет парной закрывающей метки.

**Нумерованные списки: ... **

Нумерованные списки устроены точно так же, как ненумерованные, только вместо символов, выделяющих новый элемент, используются цифры. Если слегка модифицировать наш предыдущий пример:

Добавьте к нашему примеру:

```
<OL>
<LI>клавиатура;
<LI>монитор;
<LI> мышь
</OL>
```

получится вот такой список:

1. клавиатура;
2. монитор;
3. мышь

Списки определений: <DL> ... </DL>

Список определений несколько отличается от других видов списков. Вместо меток в списках определений используются метки <DT> (от английского definition term — определяемый термин) и <DD> (от английского definition definition — определение определения). Разберем это на примере, который добавим перед меткой </body> :

```
<DL>
<DT>HTML
<DD>Термин HTML (HyperText Markup Language) означает 'язык
маркировки гипертекстов'. Первую версию HTML разработал сотрудник
Европейской лаборатории физики элементарных частиц Тим Бернерс-Ли.
<DT>HTML-документ
<DD>Текстовый файл с расширением *.html
(Unix-системы могут содержать файлы с расширением *.html).
</DL>
```

Обратите внимание: точно так же, как метки , метки <DT> и <DD> не имеют парных закрывающих меток.

Если определяемые термины достаточно коротки, можно использовать модифицированную открывающую метку `<DL COMPACT>`. Например, вот такой фрагмент HTML-текста:

```
-----  
<DL COMPACT>  
<DT>А  
<DD>Первая буква алфавита  
<DT>Б  
<DD>Вторая буква алфавита  
<DT>В  
<DD>Третья буква алфавита  
</DL>  
-----
```

Списки могут иметь атрибут **type**, но в зависимости от списка он может иметь разные параметры. Итак, параметры атрибута `type` для нумерованного списка: "A" или "a" - вместо цифр список будет нумероваться большими или маленькими латинскими буквами; "I" или "i" - список будет нумероваться большими или маленькими римскими цифрами.

Параметры атрибута `type` для ненумерованного списка: `square` - вместо точек будут квадратики; `circle` - будут незалитые кружочки, например:

`<ol type="I">` приведёт к созданию такого списка:

- I. клавиатура;
- II. монитор;
- III.мышь

а запись `<ul type="square">`, примерно такого:

- клавиатура;
- монитор;
- мышь

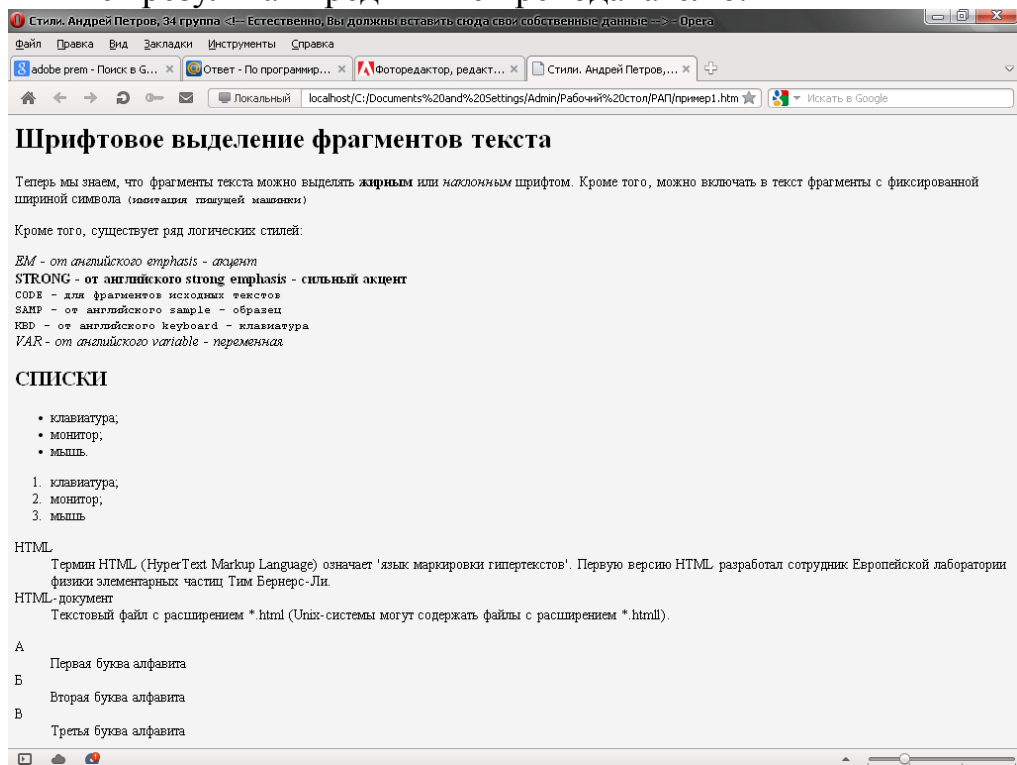
Самостоятельная работа

1. Создайте ещё 2 маркированных списка ниже имеющегося, но у этих списков в качестве маркеров должны быть окружности и квадраты соответственно.

2. Создайте ещё 4 нумерованных списка ниже имеющегося, но у этих списков в качестве номеров должны быть:

- a. большие латинские буквы;
- b. маленькие латинские буквы;
- c. большие римские цифры;
- d. маленькие римские цифры.

Получившийся результат предъявите преподавателю.



Задание 3. Списки и ссылки

Вложенные списки

Элемент любого списка может содержать в себе целый список любого вида. Число уровней вложенности в принципе не ограничено, однако злоупотреблять вложенными списками все же не следует.

Вложенные списки очень удобны при подготовке разного рода планов и оглавлений.

Наши знания о списках можно вкратце свести в пример, который нужно сохранить под именем «Виды списков.html»:

```
-----  
<html>  
<head>  
<title>  
Списки. Андрей Петров, 34 группа  
<!-- Естественно, Вы должны вставить сюда свои собственные данные -->  
</title>  
</head>  
<body>  
<H1>  
HTML поддерживает несколько видов списков  
</H1>  
<DL>  
<DT>Ненумерованные списки  
<DD>Элементы ненумерованного списка выделяются специальным  
символом и отступом слева:  
<UL>  
<LI>Элемент 1  
<LI>Элемент 2  
<LI>Элемент 3  
</UL>  
<DT>Нумерованные списки  
<DD>Элементы нумерованного списка выделяются отступом слева, а  
также нумерацией:  
<OL>  
<LI>Элемент 1  
<LI>Элемент 2  
<LI>Элемент 3  
</OL>
```

<DT>Списки определений

<DD>Этот вид списков чуть сложнее, чем два предыдущих.

<P>Помните, что списки можно встраивать один в другой, но не следует закладывать слишком много уровней вложенности.

</P>

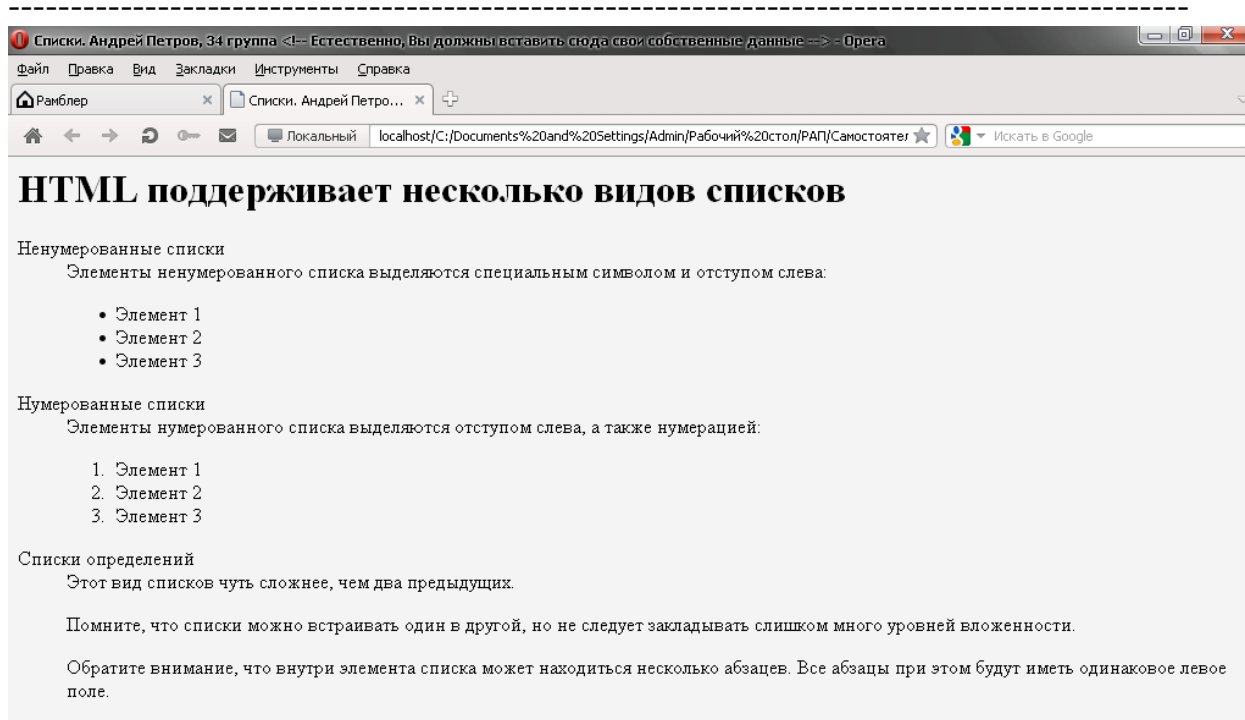
<P>Обратите внимание, что внутри элемента списка может находиться несколько абзацев. Все абзацы при этом будут иметь одинаковое левое поле.

</P>

</DL>

</body>

</html>



Форматированный текст: *<PRE> ... </PRE>*

Браузеры игнорируют множественные пробелы и символы конца строки. Из этого правила, однако, есть исключение.

Текст, заключенный между метками `<PRE>` и `</PRE>` (от английского *preformatted* — предварительно форматированный), выводится браузером на экран как есть — со всеми пробелами, символами табуляции и конца строки. Это очень удобно при создании простых таблиц.

Самостоятельное задание 1.

Выводите по центру фразу «списки определений» и вставьте между словами «списки» и «определений» 10 пробелов.

Текст с отступом: <BLOCKQUOTE> ... </BLOCKQUOTE>

Текст, заключенный между метками <BLOCKQUOTE> и </BLOCKQUOTE>, выводится браузером на экран с увеличенным левым полем.

Связывание

Как уже упоминалось, сокращение HTML означает "язык маркировки гипертекстов". Про маркировку мы уже поговорили достаточно. Не пора ли перейти к гипертексту?

Прежде всего, что же такое гипертекст? В отличие от обыкновенного текста, который можно читать только от начала к концу, гипертекст позволяет осуществлять мгновенный переход от одного фрагмента текста к другому. Системы помощи многих популярных программных продуктов устроены именно по гипертекстовому принципу. При нажатии левой кнопкой мыши на некоторый выделенный фрагмент текущего документа происходит переход к некоторому заранее назначенному документу или фрагменту документа.

В HTML переход от одного фрагмента текста к другому задается с помощью метки вида:

```
<A HREF="[адрес перехода]">  
выделенный фрагмент текста</A>
```

В качестве параметра [адрес перехода] может использоваться несколько типов аргументов. Самое простое — это задать имя другого HTML-документа, к которому нужно перейти. Например:

```
<A HREF="index.html">Перейти к новому документу</A>
```

Такой фрагмент HTML-текста приведет к появлению в документе выделенного фрагмента «Перейти к новому документу», при нажатии на который в текущее окно будет загружен документ с именем index.html.

Обратите внимание: если в адресе перехода не указан каталог, переход будет выполнен внутри текущего каталога. Если в адресе перехода не указан сервер, переход будет выполнен на текущем сервере.

Из этого следует одно очень важное практическое соображение. Если вы подготовили к публикации некоторую группу HTML-документов, которые ссылаются друг на друга только по имени файла и находятся в

одном каталоге на вашем компьютере, вся эта группа документов будет работать точно так же, если ее поместить в любой другой каталог на любом другом компьютере, в локальной сети или в Интернет! Таким образом, у Вас появляется возможность разрабатывать целые коллекции документов без подключения к Интернету, и только после окончательной готовности, подтвержденной испытаниями, помещать коллекции документов в Интернет целиком.

Рассмотрим ссылку на другой документ на примере.

Создайте новый HTML-документ в блокноте и сохраните его с именем index.html (под таким именем всегда создается стартовая страница на сайте) Текст документа представлен ниже:

```
-----  
<head>  
<title>  
Организация ссылок  
</title>  
</head>  
<body>  
<H1> Стартовая страница Андрея Петрова </H1>  
<P>  
В этом HTML-документе реализовано несколько ссылок. <BR>  
<A HREF=" Виды списков.html ">Перейти к документу с различными  
видами списков</A>  
  
</P>  
</body>  
-----
```

Посмотрите, какие изменения произошли в браузере при просмотре документа index.html. Нажмите на получившуюся ссылку.

Самостоятельное задание 2.

1. Создайте самостоятельно 3 html- документа с произвольными именами и с вашей фамилией между тегами <title> ... </title>.

В первом файле заголовком H1 оформите фразу «Нумерованный список римскими цифрами» и постройте список фамилий 4 сидящих рядом с вами студентов. В этом списке в качестве маркеров должны использоваться большие римские цифры.

Во втором файле заголовком H2 оформите фразу «Нумерованный список буквами» и постройте такой же список как и в первом файле, но в качестве маркеров должны использоваться малые латинские буквы.

В третьем файле нужно использовать заголовок H3 и маркированный список с квадратами в качестве маркеров.

2. В документе `index.html` создайте 3 ссылки на созданные вами в п.1 документы. Назовите эти ссылки «Нумерованный список1», «Нумерованный список2» и «Маркированный список».

3. Во всех созданных в 1 пункте файлах создайте ссылку с именем «Стартовая страница» на файл *`index.html`*

Получившийся результат предьявите преподавателю.

Задание 4. Вставка объектов и работа с цветовым оформлением документа

На практике часто бывает необходимо дать ссылку на документ, находящийся на другом сервере.

Введите в свой HTML-документ, созданный в предыдущем задании (index.html) такой фрагмент перед меткой `</body>`:

```
-----  
<P>  
<A HREF="http://direct-time.ru/">  
Узнайте точное время</A>  
</P>  
-----
```

Для добавления ссылки на электронную почту студенческого совета Российской академии правосудия необходимо ввести такой текст:

```
-----  
<P>  
<A HREF="mailto:sovetr@j@gmail.com "> почта студенческого совета  
</A>  
</P>  
-----
```

Изображения в HTML-документе

Встроить изображение в HTML-документ очень просто. Для этого используется тэг ``. У этого тэга есть обязательный параметр `SRC`, который указывает на файл изображения. Записывается это так:

```
<IMG SRC="picture.jpg ">
```

где `picture.jpg` – файл изображения, находящийся в одном каталоге с HTML-документом. Если каталог с изображением другой, необходимо вписать полный путь к этому файлу.

Метка `` кроме атрибута `SRC` имеет и другие, некоторые из которых приведены ниже

height=пиксели

width=пиксели

Эти атрибуты задают браузеру размер прямоугольника, в который он должен вывести картинку.

border=пиксели

Добавляет рамку к изображению

Bordercolor=цвет

Задаёт цвет рамки

hspace=пикселы

Горизонтальный отступ от изображения до окружающего контента.

vspace=пикселы

Вертикальный отступ от изображения до окружающего контента.

alt=текст

Атрибут содержит текст, который браузер выводит вместо изображения, если графика не поддерживается, или показывает в виде подсказки, когда пользователь размещает над изображением мышинный курсор. Голосовые браузеры зачитывают этот текст.

border=пикселы

Атрибут задает толщину прямоугольной рамки, которая окружает изображение. Если рамка не нужна, то используют нулевое значение: `border=0`.

align=ключ, значения ключа следующие:

left - Картинка выравнивается по левому краю, а текст обтекает картинку справа.

right - Картинка выравнивается по правому краю, а текст обтекает картинку слева.

top - Верх картинки выровнен относительно верха текущей строки.

bottom - Низ картинки выровнен по базовой линии строки (линии, на которой стоят буквы).

middle – картинка выровнена по центру текущей строки.

Разберем все, что мы знаем об изображениях с помощью примера, который впишем в наш файл `index.html` перед меткой `</BODY>`.

В качестве изображения будем использовать картинку, подобную представленной ниже, которую необходимо создать в программе Paint и сохранить в своём рабочем каталоге под именем `picture.png` (вместо этой картинки можно использовать любую другую подходящего размера):



`<H1>Изображения </H1>`

`<P>Изображение можно встроить очень просто: </P>`

`<P align=center></P>`

`<H2>Изменим размер</H2>`

`<P></P>`

`<H2>Выравнивание картинки по правому краю</H2>`

`<P>`

Ещё в 1969 году Владимир Набоков в одном из своих интервью упоминал о том, что стоит создать специальный знак пунктуации для графического отображения эмоций: "Мне часто приходит на ум, что надо придумать какой-нибудь типографический знак, обозначающий улыбку, — какую-нибудь закорючку или упавшую навзничь скобку, которой я бы мог сопроводить ответ на ваш вопрос".

`</P>`

Самостоятельная работа 1

1. В файле index.html скопируйте теги, описывающие картинку, выровненную по правому краю и следующий за ней текст. После этого, вставьте скопированный фрагмент перед меткой `</BODY>` и внесите в него изменения так, что бы обеспечить выравнивание по левому краю.

2. Поступая аналогичным образом, добавьте в файл index.html фрагменты с выравниванием:

- по центру строки;
- по верхнему краю текущей строки;
- по нижнему краю текущей строки;

3. Для одной из этих картинок необходимо задать вертикальный и горизонтальный отступ в 15 пикселей, а так же задать рамку размером 3 пикселя.

Изображение можно сделать "горячим", то есть осуществлять переход при нажатии на изображение.

Для этого, в файл index.html добавьте строки:

`<P>`

``

`</P>`

Обратите внимание на получившийся результат. Если ссылка на изображение находится между метками `` и ``, изображение фактически становится кнопкой, при нажатии на которую происходит переход по ссылке (в примере переход происходит к файлу *Виды стисков.html*).

Для того, чтобы убрать рамку у такого рисунка, необходимо добавить атрибут **border=0**

Цветовая гамма HTML-документа

Цветовая гамма всего HTML-документа определяется атрибутами, размещенными внутри метки `<BODY>`. Вот список этих атрибутов:

bgcolor - определяет цвет фона документа.

text - определяет цвет текста документа.

link - определяет цвет выделенного элемента текста, при нажатии на который происходит переход по гипертекстовой ссылке.

vlink - определяет цвет ссылки на документ, который уже был просмотрен ранее.

alink - определяет цвет ссылки в момент, когда на нее указывает курсор мыши и нажата ее правая кнопка, то есть непосредственно перед переходом по ссылке.

Цвет может кодироваться 2 способами:

1. последовательность из трех пар символов, начинающаяся с символа #. Каждая пара представляет собой шестнадцатеричное значение насыщенности заданного цвета одним из трех основных цветов (красным, зеленым и синим) в диапазоне от нуля (00) до 255 (FF).
2. Английским (американским) названием цвета.

Таблицу «безопасных» цветов можно найти в Интернет, например по адресу <http://www.artlebedev.ru/tools/colors/>, некоторые из цветов показаны ниже

Yellow	Maroon	Red	Fuchsia	Purple	Blue	Navy	Aqua	Teal	Lime	Green	Olive	White	Silver	Gray	Black
Желтый	Багровый	Красный	Малинов.	Пурпурн.	Голубой	Синий	Аква	Зел-гол.	Лайм	Зеленый	Оливковый	Белый	Серебро	Серый	Черный
#FFFF00	#800000	#FF0000	#FF00FF	#800080	#0000FF	#000080	#00FFFF	#008080	#00FF00	#008000	#808000	#FFFFFF	#C0C0C0	#808080	#000000

Разберем несколько примеров.

Если к метке `<BODY>` добавить атрибут `bgcolor` с таким сочетанием цветов:

`<BODY bgcolor=#FFFF00 >`

`bgcolor` – это цвет фона. Насыщенность красным и зеленым одинакова — FF (это шестнадцатеричное представление числа 255), а синим - 00.

Результат — желтый цвет.

Если к метке `<BODY>` добавить ещё один атрибут, который задаёт цвет текста и получим:

`<BODY bgcolor=#FFFF00 text=#0000FF >`

Насыщенность красным и зеленым одинакова — 00 (ноль), а синим — FF. Результат — синий цвет.

Если к метке `<BODY>` добавить ещё один атрибут, который задаёт цвет просмотренной гипертекстовой ссылки и получим:

`<BODY bgcolor=#FFFF00 text=#0000FF vlink=#00FF00>`

Насыщенность красным и синим — 00 (ноль), а зеленым — FF (255),. Результат — зелёный цвет ссылки.

Кроме того, метка `<BODY>` может включать атрибут **background="[имя файла]"**, который задает изображение, служащее фоном для текста и других изображений.

Браузеры заполняют множественными копиями изображения-фона все пространство окна, в котором открыт документ, подобно тому, как при строительстве большие пространства стен покрывают маленькими (и одинаковыми) плитками.

Самостоятельная работа 2

1. Раскрасьте все элементы созданного вами сайта в произвольные цвета (проявление дизайнерских способностей приветствуется).

Цвета должны быть заданы для:

- Фона
 - Текста
 - Просмотренных ссылок
 - Не просмотренных ссылок
2. Найдите в Интернете произвольный рисунок и сделайте его фоновым изображением своего сайта.

Предъявите задание преподавателю.

Задание 5. Создание таблиц, вставка звука и видео

Как устроена таблица

В устройстве таблицы легче всего разобраться на простом примере, текст которого нужно сохранить под именем Таблица.html:

`<html>`
`<head>`

```

<title>
Таблицы. Андрей Петров, 34 группа
<!-- Естественно, Вы должны вставить сюда свои собственные данные -->
</title>
</head>
<body>
<H1>
Простейшая таблица
</H1>
<TABLE BORDER=1> <!--Это начало таблицы-->
<CAPTION> <!--Это заголовок таблицы-->
У таблицы может быть заголовок
</CAPTION>
  <TR> <!--Это начало первой строки-->
    <TD> <!--Это начало первой ячейки-->
    Первая строка, первая колонка
    </TD> <!--Это конец первой ячейки-->
    <TD> <!--Это начало второй ячейки-->
    Первая строка, вторая колонка
    </TD> <!--Это конец второй ячейки-->
  </TR> <!--Это конец первой строки-->
  <TR> <!--Это начало второй строки-->
    <TD> <!--Это начало первой ячейки-->
    Вторая строка, первая колонка
    </TD> <!--Это конец первой ячейки-->
    <TD> <!--Это начало второй ячейки-->
    Вторая строка, вторая колонка
    </TD> <!--Это конец второй ячейки-->
  </TR> <!--Это конец второй строки-->
</TABLE> <!--Это конец таблицы-->
</body>
</html>

```

Таблица начинается с метки <TABLE> и заканчивается меткой </TABLE>. Метка <TABLE> может включать несколько атрибутов:

ALIGN

Устанавливает расположение таблицы по отношению к полям документа. Допустимые значения: ALIGN=LEFT (выравнивание влево), ALIGN=CENTER (выравнивание по центру), ALIGN=RIGHT (выравнивание вправо).

WIDTH

Ширина таблицы. Ее можно задать в пикселах (например, WIDTH=400) или в процентах от ширины страницы (например, WIDTH=80%).

BORDER

Устанавливает ширину внешней рамки таблицы и ячеек в пикселах (например, BORDER=4). Если атрибут не установлен, таблица показывается без рамки.

CELLSPACING

Устанавливает расстояние между рамками ячеек таблицы в пикселах (например, CELLSPACING=2).

CELLPADDING

Устанавливает расстояние между рамкой ячейки и текстом в пикселах (например, CELLPADDING=10).

Таблица может иметь заголовки (<CAPTION> ... </CAPTION>), хотя заголовок не является обязательным. Метка <CAPTION> может включать атрибут ALIGN. Допустимые значения: <CAPTION ALIGN=TOP> (заголовок помещается над таблицей) и <CAPTION ALIGN=BOTTOM> (заголовок помещается под таблицей).

Каждая строка таблицы начинается с метки <TR> и заканчивается меткой </TR>. Метка <TR> может включать следующие атрибуты:
ALIGN

Устанавливает выравнивание текста в ячейках строки. Допустимые значения: ALIGN=LEFT (выравнивание влево), ALIGN=CENTER (выравнивание по центру), ALIGN=RIGHT (выравнивание вправо).

VALIGN

Устанавливает вертикальное выравнивание текста в ячейках строки. Допустимые значения: VALIGN=TOP (выравнивание по верхнему краю), VALIGN=MIDDLE (выравнивание по центру), VALIGN=BOTTOM (выравнивание по нижнему краю).

Каждая ячейка таблицы начинается с метки <TD> и заканчивается меткой </TD>. Метка <TD> может включать следующие атрибуты:

NOWRAP

Присутствие этого атрибута означает, что содержимое ячейки должно быть показано в одну строку.

COLSPAN

Устанавливает "размах" ячейки по горизонтали. Например, COLSPAN=3 означает, что ячейка простирается на три колонки.

ROWSPAN

Устанавливает "размах" ячейки по вертикали. Например, ROWSPAN=2 означает, что ячейка занимает две строки.

ALIGN

Устанавливает выравнивание текста в ячейке. Допустимые значения: ALIGN=LEFT (выравнивание влево), ALIGN=CENTER (выравнивание по центру), ALIGN=RIGHT (выравнивание вправо).

VALIGN

Устанавливает вертикальное выравнивание текста в ячейке. Допустимые значения: VALIGN=TOP (выравнивание по верхнему краю), VALIGN=MIDDLE (выравнивание по центру), VALIGN=BOTTOM (выравнивание по нижнему краю).

WIDTH

Устанавливает ширину ячейки в пикселах (например, WIDTH=200).

HEIGHT

Устанавливает высоту ячейки в пикселах (например, HEIGHT=40).

Если ячейка таблицы пуста, вокруг нее не рисуется рамка. Если ячейка пуста, а рамка нужна, в ячейку можно ввести символьный объект (non-breaking space — неразрывный пробел). Ячейка по-прежнему будет пустой, а рамка вокруг нее будет.

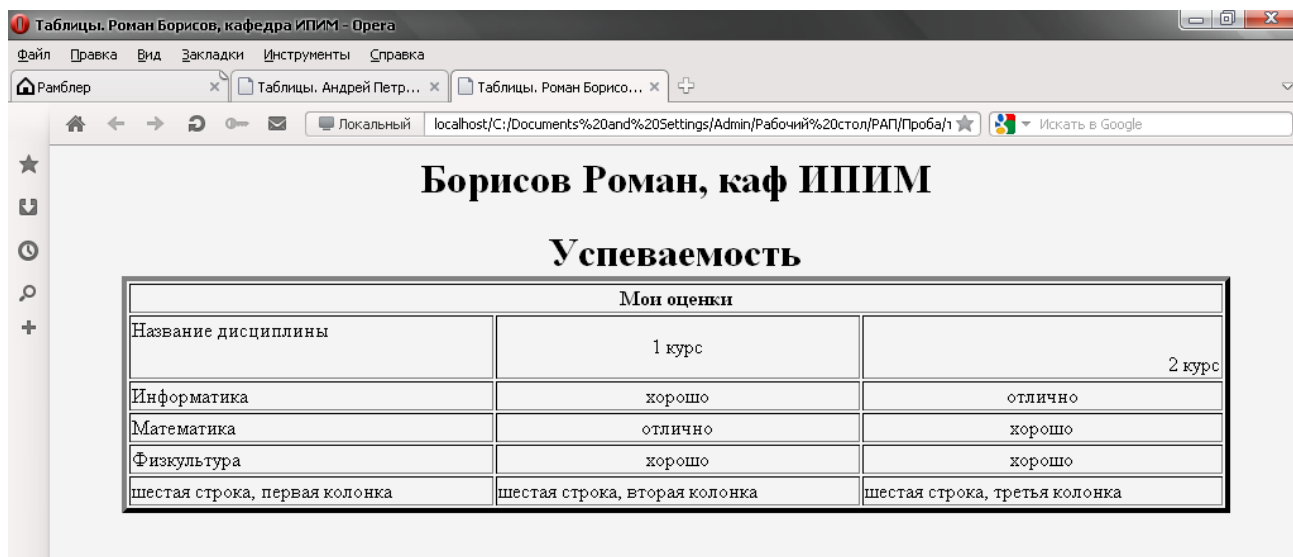
Самостоятельная работа 1

1. В заголовок документа Таблица.html введите свою фамилию, имя, номер группы, выровняйте заголовок по центру.
2. Отредактируйте существующую таблицу установив в ней 6 строк и 3 столбца.

Совет: используйте комментарии, что бы в последствии «не заблудиться» в получившейся таблице.

3. Выберите выравнивание таблицы по центру, ширина таблицы – 90% от экрана, задайте рамку толщиной 4 пиксела.
4. Объедините верхние 3 ячейки таблицы, для чего оставьте в 1 строке только 1 ячейку и с помощью атрибута COLSPAN установите ее размер в 3 столбца.
5. Высоту 2 строки установите в 50 пикселей и в этой строке установите следующие параметры выравнивания: 1 ячейка по левому краю и по верху, 2 ячейка по центру по горизонтали и вертикали, 3 ячейка – по правому краю и по низу.
6. Введите в таблицу текст, оформив его как на образце и проставьте свои оценки по предметам.

В результате должна получиться таблица, похожая на представленную на рисунке



Вставка звуковых и видео файлов

Для внедрения звукового файла во время просмотра сайта необходимо ввести следующую строку:

```
<bgsound src="url/name.wav" loop="2">
```

где url/name.wav – путь и имя звукового файла,

loop="2" указание проиграть файл 2 раза

либо можно использовать такую строку:

```
<embed src="url/name.wav">
```

В этом случае произойдёт внедрение звукового файла, а на экране появляется панель управления проигрыванием звука

Для создания ссылки на музыкальный файл необходимо набрать следующую строку.

```
<a href="url/name.wav">Прослушать музыку</a>
```

где url/name.wav – путь и имя звукового файла,

прослушать музыку – текст, который будет оформлен в виде

гиперссылки.

Видеофайл внедряется аналогичным образом:

<embed src="url/name.avi"> - внедрение видеофайла с панелью управления проигрыванием видео

 - внедрение видеофайла без панели управления, видео будет проиграно один раз

Посмотреть видео - внедрение видеофайла по ссылке

Самостоятельная работа 2

Найдите в Интернете 3 видеоролика, длительностью не более 1 минуты и разместите ссылки на них в шестой строке созданной таблицы, выровняв все ссылки по центру ячейки. Обратите внимание, все ссылки должны быть рабочими.

Результат предъявите преподавателю.

Задание 6. Контрольная работа по созданию html документа

Выполнение контрольной работы по дисциплине заключается в создании WEB-сайта на основе знаний и навыков, полученных на предыдущих занятиях по дисциплине.

Сайт создается последовательно по этапам, требования к которым изложены ниже.

1. Создать файл index.html. При открытии этого файла в браузере на странице должна быть отображена информация:

- ФИО автора сайта, номер учебной группы, номер варианта задания, тема контрольной работы. Все перечисленные элементы должны быть оформлены различными типами заголовков и выровнены по центру.
- Ссылки на будущие страницы создаваемого сайта (не менее двух).

У страницы должен быть фоновый рисунок, выбранный автором самостоятельно, другие элементы оформления – по желанию.

2. Подобрать исходный материал с иллюстрациями для оформления WEB-страницы. Материал выбирается согласно варианта из перечня, размещённого ниже, объёмом не менее трёх электронных страниц. По согласованию с преподавателем можно выбрать тему, не входящую в перечень.

На основе подобранного материала создать одну или несколько страниц, на каждой из которых должна быть ссылка на главную страницу (index.html). Страница должна иметь фон, иллюстрации с различными способами выравнивания.

3. Создать страницу "Об авторе" и поместить на нее:

- Нумерованный список (например, список своих увлечений)
- Маркированный список (например, список своих друзей)
- Таблицу с заголовком и содержимым размером не менее 5x5 (например, с дисциплинами текущего семестра, фамилиями преподавателей и текущими оценками)
- Ссылки на аудио и видео файлы (объёмом не более 10 Мбайт)
- Фото автора.
- Ссылку на адрес электронной почты автора
- Оформить страницу с применением способов, изученных на предыдущих занятиях.

Оформление курсовой работы

Работа представляется в электронном виде в формате *.html

Защита контрольной работы

При защите работы студент должен знать:

1. Материал контента своего сайта
2. Методы и средства создания html документов, использованных в работе.

Приложение: Список тем для контента страниц по вариантам

Темы по информационным системам и технологиям

1. Форматы текстовых файлов.
2. Графические файлы. Понятие растровой графики
3. Графические файлы. Понятие векторной графики.
4. Цветовая модель RGB.
5. Цветовая модель SMYK.
6. Форматы графических файлов.
7. Форматы звуковых файлов.
8. Форматы сохранения видеoinформации.
9. Сжатие видеoinформации.
10. Типы дисплеев.
11. ЭЛТ и ЖК дисплеи.
12. Плазменные дисплеи, ELD, FED, LEP дисплеи.
13. Типы акустических систем.
14. Типы устройств виртуальной реальности.
15. Диски, приводы, стандарты записи на дисках.
16. Общие положения синтеза звука. FM и WT синтез.
17. Основные типы современных видеокарт.
18. Основные типы современных звуковых карт.
19. Основные типы современных TV- тюнеров.
20. Основные типы современных видеопроекторов.
21. Принципы построения трехмерного изображения.
22. Принципы формирования объемного звука.
23. Основные виды современных печатающих устройств.
24. Основные принципы сканирования изображений.
25. Основные типы современных цифровых видеокамер.
26. Основные типы и характеристики современных цифровых фотоаппаратов.
27. Основные типы и характеристики сенсорных дисплеев и панелей.
28. Основные типы и характеристики современных PDA устройств.
29. Мультимедиа-технологии.
30. Программы для офисной автоматизации.
31. Области применения искусственного интеллекта.
32. Обзор существующих экспертных систем.
33. Примеры решения типовых задач управления производством на ЭВМ.

34. Обзор существующих автоматизированных обучающих систем.
35. АСНИ – средство повышения эффективности исследования.
36. Кодирование графической информации в компьютере.
37. Кодирование звука в компьютере.
38. Преобразование аналогового сигнала в цифровой. Основные принципы и этапы.
39. Классификация компьютеров.
40. Основные узлы персонального компьютера и основные виды периферийного оборудования.
41. Основные виды информационных технологий и их краткая характеристика.
42. Технологии телекоммуникационного доступа

Темы по праву и организации социального обеспечения

1. Понятие и виды социального обеспечения. Функции социального обеспечения.
2. Понятие, предмет, метод и система права социального обеспечения.
3. Основные принципы права социального обеспечения.
4. Источники права социального обеспечения.
5. Правоотношения по социальному обеспечению.
6. Финансирование социального обеспечения. Организационные и правовые основы обязательного пенсионного страхования.
7. История развития законодательства о социальном обеспечении.
8. Понятия, виды и значение трудового стажа в социальном обеспечении. Исчисление общего трудового стажа, используемого в целях оценки пенсионных прав застрахованных лиц. Виды деятельности, включаемые в страховой стаж. Правила исчисления страхового стажа.
9. Понятие и значение стажа на соответствующих видах работ. Виды стажа на соответствующих видах работ. Общая характеристика списков соответствующих видов работ.
10. Досрочные трудовые пенсии по старости.
11. Виды трудовых пенсий. Понятие трудовой пенсии по старости. Структура трудовых пенсий.
12. Пенсии по ГПО. Виды пенсий. Структура пенсий по ГПО.
13. Категории граждан, подвергшихся радиации. Правовое регулирование их социальной защиты. Категории граждан, пострадавших вследствие Чернобыльской катастрофы. Зоны радиоактивного заражения.
14. Федеральные государственные гражданские служащие.
15. Пособия. Виды пособий и их классификация.
16. Государственная социальная помощь (понятие, цель оказания, виды, исчисление среднедушевого дохода, размер).
17. Порядок предоставления государственной социальной помощи в форме ежемесячных денежных выплат.
18. Учреждения стационарной формы социального обслуживания. Организация содержания и обслуживания престарелых и инвалидов в домах-интернатах.
19. Организация социального обслуживания граждан пожилого возраста и инвалидов через отделения социального обслуживания на дому (в т.ч. социально-медицинское обслуживание).
20. Комплексные Центры социального обслуживания населения (порядок образования, структура). Служба срочной социальной помощи. Отделения дневного пребывания (цель создания, основные направления деятельности).

21. Профессиональное обучение и трудоустройство инвалидов.
22. Организация протезирования инвалидов и обеспечение средствами передвижения.
23. Понятие социального обеспечения и социального страхования.
24. Инвалидность и социальная защита инвалидов в РФ.
25. Трудовой стаж и его значение в социальном обеспечении.
26. Общая характеристика системы пенсионного обеспечения в РФ.
27. Пособия в праве социального обеспечения.
28. Пособие по временной нетрудоспособности.
29. Пособие по безработице.
30. Обеспечение в связи с несчастными случаями на производстве и профессиональными заболеваниями.
31. Государственные пособия гражданам, имеющим детей.
32. Льготы и компенсации в праве социального обеспечения.
33. Медицинская помощь и лечение.
34. Правовое регулирование социального обслуживания населения в РФ.