# Лабораторная работа №2 «Язык гипертекстовой разметки HTML»

## 1.1. Теоретические основы

Работу языка HTML можно сравнить с текстовым редактором (например, MS Word). С помощью MS Word обычный текст можно превратить в грамотно оформленный документ, содержащий абзацы, списки, курсив, а также изображения.

Основу разметки HTML составляют «теги» - специальные команды, описывающие структуру и формирующие содержимое web-страницы. Теги предоставляют браузеру информацию о форматировании и разметке страницы [3].

Название тега заключается в угловые скобки « < » и « > ». Некоторые теги могут включать в себя атрибуты, с помощью которых можно форматировать их содержимое.

Теги можно разделить на следующие категории: одиночные и парные.

Примеры одиночных тегов:

<img> - позволяет добавить в документ изображение

<br> - тег переноса строки

<input> - элемент формы (текстовое поле, флажок, кнопка)

Примеры парных тегов:

<h1>...</h1> - создаёт заголовок первого уровня (самый крупный)

<p>...</p> - позволяет создать абзац

<b>...</b> - задает жирное начертание шрифта

Среди многообразия тегов можно выделить главные теги (верхнего уровня), которые определяют структуру web-документа, и всегда должны быть прописаны в каждой html-странице.

**<html>…</html>** - указывает браузеру, что он имеет дело с HTML-документом.

**<head>…</head>** - содержит описание web-страницы и является контейнером для всех заголовочных элементов html-документа, цель которых - помогать браузерам и поисковым системам в работе с данными.

**<body>…</body>** - определяет видимую часть web-документа.

Структура web-документа представляет собой иерархию, подчиняющуюся определенным правилам.

## 1.2. Структура web-документа

Любой web-документ подчиняется единым правилам структурной организации. Общая схема структуры web-документа представлена на рис. 1.1.



Рисунок 1.1. Общая схема структуры web-документа

Тег <DOCTYPE> отвечает за корректное отображение web-страницы браузером. DOCTYPE определяет не только версию HTML (например, html), но и соответствующий DTD-файл в Интернете.

DTD (Document Type Definition — определение типа документа) - термин, который используется для описания схемы документа или его части языком схем DTD и представляет собой XML-документ, определяющий, какие теги, атрибуты и их значения действительны для конкретного типа HTML. Для каждой версии HTML есть свой DTD.

Тег <HTML> является корневым элементом документа, а также контейнером для всех остальных элементов. Все, что находится за пределами тегов <html>…</html>, не воспринимается браузером как код HTML и никак им не обрабатывается.

Раздел <HEAD>…</HEAD> содержит техническую информацию о странице: заголовок, ключевые слова для поисковых машин, кодировку.

В разделе <BODY>…</BODY> располагается содержимое web-документа.

Любой документ в файловой системе имеет своё расширение: doc, xls, txt, pdf, благодаря ему файл распознаётся специальными программами-интерпретаторами (MS Word, MS Excel, Блокнот, Adobe Reader). Аналогично html-документ должен иметь собственное расширение: html или htm.

С документами в формате html можно работать в различных средах, начиная с встроенного в Windows Блокнота, заканчивая специализированными текстовыми HTML - редакторами.

## 1.3. Специализированные редакторы HTML-кода

Специализированные редакторы html-кода предоставляют пользователю следующие возможности:

* подсветка синтаксиса
* встроенные справочники
* проверка документа на ошибки
* встроенный браузер

HTML-редакторы делятся на две категории: графические и программные редакторы. Внешне редакторы обоих типов выглядят очень похожими. Различия заключаются в методах визуального представления элементов, составляющих Web-страницу.

Графические редакторы представляют страницу такой, какой она будет в окне браузера. Программа (код HTML) встроена в страницу, но скрыта от автора, который не работает с ней напрямую.

Программные редакторы выводят на экран в качестве основного представления страницы исходный текст на языке HTML, предоставляя при этом в распоряжение автора мощные средства генерации кода, избавляющие от необходимости писать его вручную.

В графических и программных редакторах предлагаются похожие средства создания и редактирования страниц. Кнопки инструментальной панели дают возможность быстрого доступа к таким элементам, как изображения, списки и связи, или к кодам, служащим для создания и размещения этих элементов на странице; как правило, они вызывают диалоговые окна, которые помогают на всех этапах проектирования. Многие продукты содержат "мастеров", автоматизирующих, по крайней мере, базовые процедуры создания страниц и способных помочь при построении форм, навигационных карт и выполнении других сложных операций.

Таким образом, хороший редактор HTML-кода это такой редактор, который способен максимально избавить пользователя от непосредственного написания HTML-кода, при этом оставляя ему свободу выбора и код "на выходе" не должен содержать ошибок и разного "мусора", очистка от которого может занять много времени.

Среди таких редакторов можно выделить несколько:

1. **Notepad++**. Формально не является редактором HTML, однако активно используется для этих целей. В него встроена подсветка синтаксиса языков разметки HTML и XML, большинства популярных языков программирования, часть которых используется и в web-разработке, такие как PHP, Ruby, Perl, Python и множество других.
2. **WebCoder -** простой редактор HTML, PHP и ASP кода.  Небольшая, бесплатная программа, которая пригодится для создания и редактирования веб-страниц.  С ее помощью можно редактировать исходный код, производить поиск и замену в коде, вставлять теги HTML и просматривать как будет выглядеть страница в браузере. Поддерживается функция Wordwrap, подсчет количества строк и веса страницы, возможность изменить регистр выделенного кода или всего документа, вставка даты в разных форматах и т.п.
3. **Sublime Text**. Платный текстовый редактор (условно-платный).

При выполнении заданий пособия будет использоваться редактор **Notepad++**.

Если он не установлен на вашем компьютере, следует скачать его в Интернете и установить.

## 1.4. Организация рабочих файлов проекта

При создании нового проекта (web-приложения, web-сайта) следует организовать рабочее пространство. С практической точки зрения все файлы проекта следует хранить в одной директории (папке). К файлам проекта относятся: html-страницы (\*.html), файлы стилей (\*.css), сценарии JavaScript (\*.js), изображения (\*.jpg, \*.png) , php-файлы (\*.php).

|  |
| --- |
| *Задание.*  В предыдущей лабораторной работе должна была быть создана папка проекта **UCHET\_FIO** (например, **UCHET\_IvanovII**). В ней создадим файл *index.php*. Дополнительно в той же директории создайте папки с именами «css», «js» и «img». По степени нарастания сложности приложения, в данных папках будут располагаться соответствующие файлы. |

|  |
| --- |
| *Примечание.*  По умолчанию нельзя создать файл с расширением *php*. Создайте **текстовый документ**, откройте его, выберите пункт меню *Файл → Сохранить как* и выполните приведенные ниже настройки (см. рис. 1.2). Не забудьте указать кодировку файла UTF-8. |

Рисунок 1.2. Действия по созданию файла с расширением php

В результате содержимое рабочей папки проекта будет выглядеть следующим образом (см. рис. 1.3).

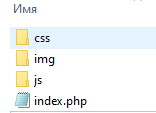
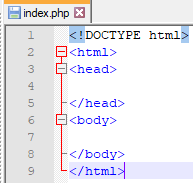


Рисунок 1.3. Структура папки с проектом

Следующим шагом начнём формирование структуры документа.

## 1.5.Формирование структуры документа

После выполнения описанных выше операций открываем файл *index.php* с помощью Notepad++ или другого редактора кода. Формируем рассмотренную выше структуру (см. рис. 1.1) web-документа.



Если открыть данную страницу в браузере, написанный в файле код отобразится в том же виде (см. рис. 1.4).

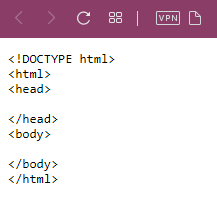


Рисунок 1.4. Отображение файла index.php в браузере

Это связано с тем, что для обработки кода php-файлов необходимо, чтобы на компьютере был запущен PHP-интерпретатор. Его запуск осуществляется в момент запуска EasyPHP.

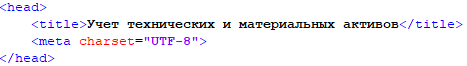
Начнём с создания заголовка страницы. Он создаётся в разделе *<HEAD>…</HEAD>* с помощью тегов *<title>…</title>*.

Также, при формировании файла следует кодировку его содержимого. Это позволит избежать проблем с отображением web-страницы в браузере.

Для указания кодировки в HTML используется специальный тег *<meta>*.

*<meta charset=”UTF-8”>*

Вносим в файл описанные выше изменения.



Далее перейдём к формированию содержимого главной страницы приложения.

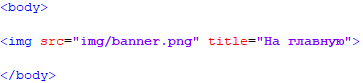
## 1.6. Формирование главной страницы приложения

Главная страница приложения будет содержать баннер-картинку, главное меню для навигации по сайту и контент, содержащий информацию о специфике выбранной предметной области.

Картинка для баннера была подготовлена заранее и помещена в папке **img** под именем «*banner.png*».

|  |
| --- |
| *Примечание.*  Все упоминаемые в пособии дополнительные файлы (изображения, библиотеки и т.д.) находятся в папке «*Файлы для выполнения МУ*». |

Загрузим картинку на страницу с помощью тега *<img>*.



Атрибут *title* задаёт подсказку при наведении курсора мыши на изображение.

|  |
| --- |
| *Задание.*  Перенесите файл *banner.png* в папку **img** в своей рабочей директории и подключите его к странице в файле *index.php* в самое начало секции *<body>…</body>*. |

Следует обратить внимание на атрибут *src*, который содержит путь к изображению. Так как все изображения в проекте будут храниться в папке **img**, то перед файлом необходимо указывать имя папки.

Тег **<img>** является одиночным тегом, следовательно, закрывать его не требуется.

Также в приложении будет предусмотрена возможность регистрации/авторизации пользователя.

Для этого добавим на страницу ссылки для регистрации и авторизации пользователя в системе (см. рис. 1.5).



Рисунок 1.5. Добавление в файл ссылок для регистрации/авторизации

Следующим шагом сформируем главное меню приложения. Для этого будем использовать списки. Для создания меню потребуется использовать вложенные списки (см. рис. 1.6).

Следует отметить, что список сам по себе не может представлять меню сайта или приложения. Каждый элемент списка должен служить ссылкой на соответствующую страницу.

В данном случае используются относительные гиперссылки. Каждый пункт меню ссылается на определённый файл, лежащий в той же директории, что и *index.php*.

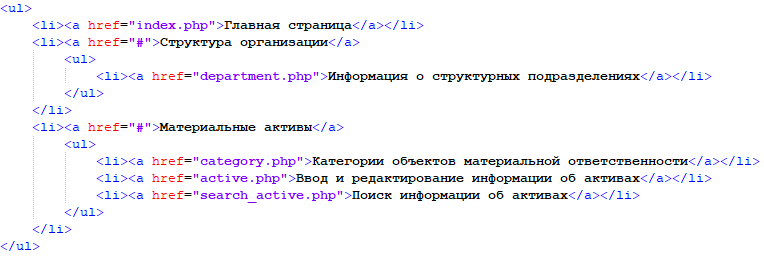


Рисунок 1.6. Формирование главного меню приложения

|  |
| --- |
| *Задание.*  Добавьте на страницу список с указанием ссылок на соответствующие страницы.  Создайте в своей рабочей папке файлы *reg\_form.php*, *auth\_form.php*, *department.php*, *category.php*, *active.php* и *search\_active.php*. Они пригодятся в будущем. Файлы должны быть созданы в кодировке UTF-8. |

Можно заметить, что пункт меню «**Главная страница**» ссылается на файл *index.php*. Главное меню будет дублироваться на каждой странице приложения и из любого места приложения пользователь сможет вернуться на главную страницу. Дополнительно сделаем ссылкой картинку-баннер, чтобы при нажатии на неё происходил переход на главную страницу приложения.



Следующим шагом добавим на страницу вводную информацию о задачах приложения. Оформим её в виде таблицы с применением тегов форматирования текста (см. рис. 1.7).

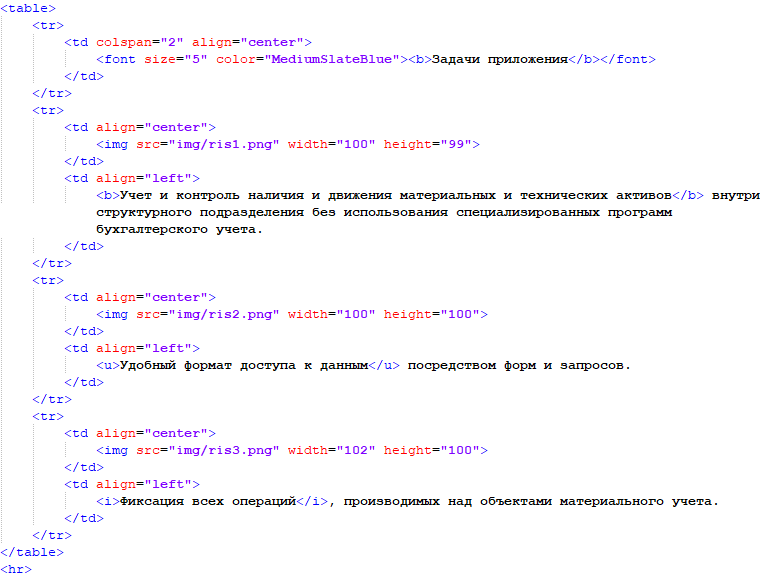


Рисунок 1.7. Код для формирования вводной информации на главной странице приложения

|  |
| --- |
| *Примечание.*  Для оформления текста используются такие теги как *<b>*, *<i>*, *<u>*, *<font>*.  Теги форматирования текста приведены в качестве справки. Основное форматирование и стилизация приложения будет производиться средствами CSS.  Тег **<hr>** создаёт горизонтальную линию, визуально отделяя друг от друга элементы главной страницы приложения. |

|  |
| --- |
| *Примечание.*  Файлы «*ris1.png*», «*ris2.png*», «*ris3.png*» необходимо заранее скопировать из папки «*Файлы для выполнения МУ*» в папку **img** в своей рабочей директории. |

Код данного блока приведен ниже:

|  |
| --- |
| <hr>  <table>  <tr>  <td colspan="2" align="center">  <font size="5" color="MediumSlateBlue"><b>Задачи приложения</b></font>  </td>  </tr>  <tr>  <td align="center">  <img src="img/ris1.png" width="100" height="99">  </td>  <td align="left">  <b>Учет и контроль наличия и движения материальных и технических активов</b> внутри структурного подразделения без использования специализированных программ бухгалтерского учета.  </td>  </tr>  <tr>  <td align="center">  <img src="img/ris2.png" width="100" height="100">  </td>  <td align="left">  <u>Удобный формат доступа к данным</u> посредством форм и запросов.  </td>  </tr>  <tr>  <td align="center">  <img src="img/ris3.png" width="102" height="100">  </td>  <td align="left">  <i>Фиксация всех операций</i>, производимых над объектами материального учета.  </td>  </tr>  </table>  <hr> |

|  |
| --- |
| *Задание.*  Добавьте на страницу после секции с будущим меню указанный выше код. Запустите проект и проверьте корректность отображения всех элементов. |

|  |
| --- |
| *Примечание.*  Для запуска проекта необходимо:  - запустить EasyPHP  - убедиться, что оба светофора зелёные (Apache и MySQL успешно запущены)    - в адресной строке браузера ввести 127.0.0.1 (или localhost)  - найти в списке проектов свой проект (*UCHET\_IvanovII*) и нажать на него    Если открыть файл *index.php* в браузере обычным способом, на странице отобразится программный код. |

В результате выполненных действий страница будет выглядеть следующим образом (см. рис. 1.8).

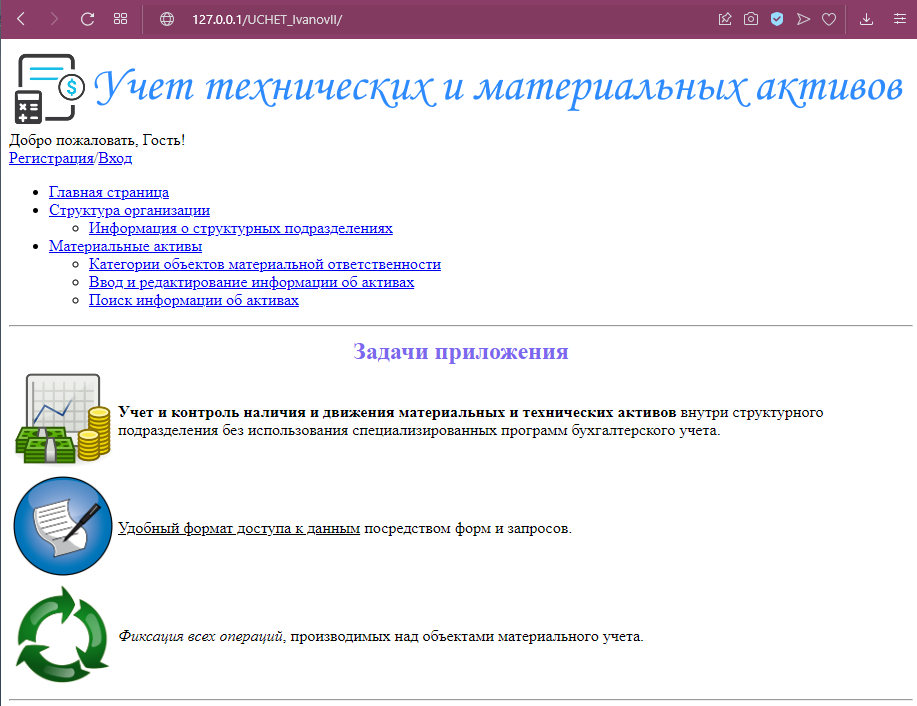


Рисунок 1.8. Отображение главной страницы приложения в браузере

## 1.7. Добавление иконки сайта для вкладки в браузере

По умолчанию у вкладки приложения в браузере нет иконки. Для её добавления можно добавить следующую строчку кода в секцию *<head>…</head>*:

<link rel="icon" href="img/icon.png" type="image/png">

Результат добавления иконки виден непосредственно на вкладке в браузере:



|  |
| --- |
| *Задание.*  Внесите в файл *index.php* указанную строчку кода. Проверьте результат. |

В результате выполнения описанных выше процедур структура рабочей папки будет выглядеть следующим образом (см. рис. 1.9).

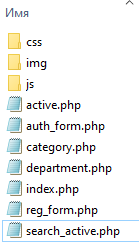


Рисунок 1.9. Структура рабочей директории с файлами

Файлы с расширением *\*.php* на текущий момент пустые. Для дальнейшей работы следует изучить основы каскадных таблиц стилей CSS для оформления приложения.