

Лабораторная работа № 1

Тема “ Структура HTML – документа”.

Цель работы: Основные правила написания страницы на HTML.

HTML – язык гипертекстовой разметки. Данный формат определяет внешний вид документа, взаимное расположение текстовой, графической и мультимедийной информации.

Основу html-документа составляет текстовый документ, который можно набрать в любом текстовом редакторе. Созданный файл нужно сохранить с расширением *.html.

Любой документ, написанный на html, состоит из текста – содержимого страницы и управляющих символов – тегов, определяющих расположение и форматирование текста.

Основные теги:

Весь код страницы помещается внутри тегов <html>..</html>

Html-документ, как правило, имеет заголовок, размещенный в тегах <head>..</head>

В заголовке может быть указан титул документа, т.е. его название в заголовке окна браузера. Титул заключается в теги <title>..</title>

Также в заголовке часто указывается дополнительная информация в теге <meta> (кодировка, ключевые слова для поисковых машин, информация об авторе и т.д.). Например, тег <meta http-equiv="content-type" content="text/html; charset=windows-1251"> предназначен для определения кодировки, с помощью которой была написана страница.

Содержимое страницы заключается в теги <body>..</body>

Пример

```
<html>
<head>
  <meta http-equiv="content-type" content="text/html; charset=windows-1251">
  <title>
    Это html-документ
  </title>
</head>
<body>
  Это моя первая страница
</body>
</html>
```

Задание

Скопируйте приведенный выше код в текстовый редактор Блокнот. Сохраните страницу с расширением html. Для этого поле «Тип файла» (1) укажите «Все файлы», в поле «Имя файла» (2) - имя_файла.html (рис.1). Привыкайте давать страницам латинские названия, желательно еще не использовать пробелы. Обычно главную страницу сайта называют index.html.



Рис.1 Сохранение страницы

Страница, соответствующая примеру, выглядит следующим образом (рис.2):

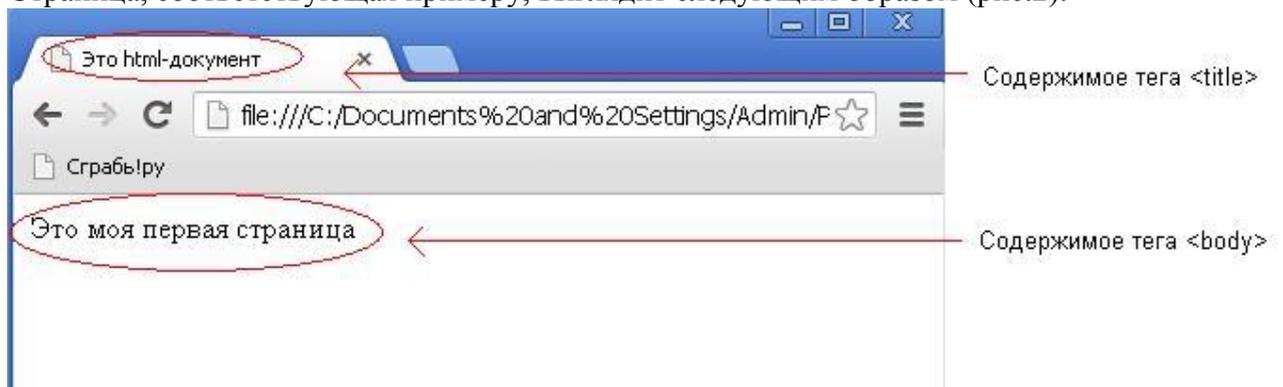


Рис.2 Страница в браузере

Для внесения изменений в уже созданную страницу, необходимо открыть ее с помощью текстового редактора (рис.3).

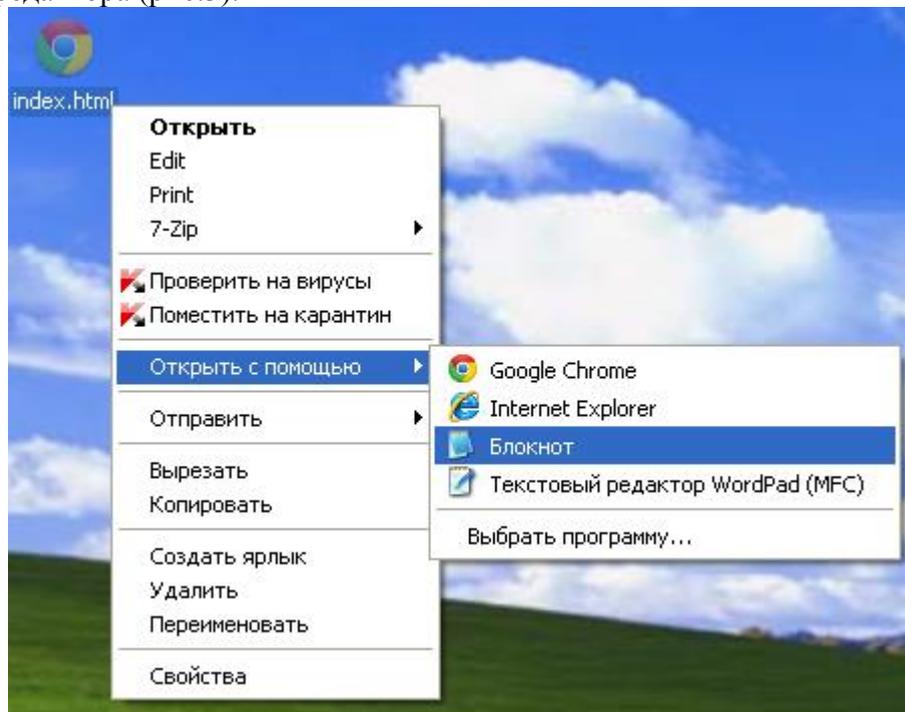


Рис.3 Открытие страницы в текстовом редакторе

Атрибуты тега <body>:

Тег <body> имеет ряд необязательных атрибутов. Атрибуты записываются внутри открывающего тега через пробел:

background – фоновое изображение, на котором отображается текст документа.

Пример:

`<body background="fish.jpg">`- относительная адресация, изображение должно быть сохранено там же, где и html-страница.

`<body background="images/fish.jpg">` - относительная адресация, изображение должно быть сохранено в папке images, папка - там же, где и html-страница (наиболее удобный способ).

`<body background="c:/images /fish.jpg">` - абсолютная адресация, изображение должно быть сохранено по адресу c:/images, независимо от места хранения html-страницы.

bgcolor – задает цвет фона.

Пример:

`<body bgcolor=red>` - содержимое страницы будет отображаться на красном фоне.

`<body bgcolor=#FF0000>` - содержимое страницы будет отображаться на красном фоне.

Цвет можно указать либо в виде строковой константы (узнать их можно в файле Дополнительный материалы/Кодировка цвета.doc), либо в виде 6-разрядного 16-ричного числа, каждые два разряда которого определяют цвет палитры RGB (#FF0000 – красный, #00FF00 – зеленый, #0000FF – синий, #000000 – черный, #FFFFFF – белый и т.д.).

text – задает цвет текста.

link – задает цвет гиперссылки (по умолчанию – синий).

vlink– задает цвет пройденной гиперссылки (по умолчанию – фиолетовый).

Задание

Создайте html-страницу, соответствующую образцу «Образец.jpg» из папки «Задание1». В качестве фона используйте файл key.jpg.

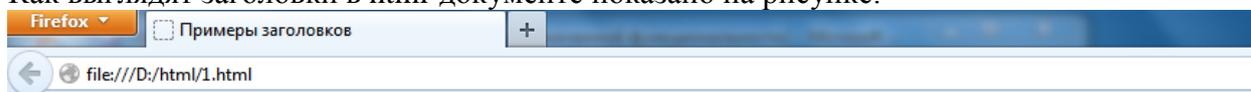
Лабораторная работа № 2

Тема «Заголовки и форматирование».

Цель работы: Изучение правил написания заголовков.

Заголовок означает начало раздела документа. Определены 6 уровней заголовков – от `<h1>..</h1>` (самый крупный) до `<h6>..</h6>` (самый мелкий).

Как выглядят заголовки в html-документе показано на рисунке:



Заголовок 1. Это может быть название главы или раздела

Заголовок 2. Это может быть название параграфа

Заголовок 3. Это может быть название подпункта

Заголовок 4. Это может быть название еще более мелкого фрагмента

Заголовок 5. Это может быть название еще более мелкого фрагмента

Заголовок 6. Это может быть название еще более мелкого фрагмента

Рис.4 Внешний вид заголовков в html

.. - тег изменения шрифта.

Атрибуты тега ****:

face – тип шрифта.

color – цвет шрифта.

size – размер шрифта.

В следующем примере слово «Текст» выделяется красным цветом, шрифт – Arial, размер 10.

Пример

```
<font face=Arial color=red size=10> Текст </font>
```

<p> - тег вставки нового абзаца. Этот тег можно не закрывать, т.е. не использовать команду **</p>**, в этом случае признаком окончания абзаца будет тег **<p>**, открывающий новый абзац.

Атрибуты тега **<p>**:

align - выравнивание текста внутри абзаца. Возможны значения **left, right, center**.

В следующем примере абзац со словом «Текст» выравнивается по правому краю.

Пример

```
<p align=right> Текст </p>
```

**
** - используется для принудительного переноса текста на новую строку.

Визуально между абзацами есть интервал, между строками, разделенными **
** интервала нет (см. рисунок).

"...Знающим я себя назвать не смел.
Я был ищущим и все еще остаюсь им...
Г.Гессе, "Демриан"

Здесь для перехода на новую строку использовался тег **
**

Здесь для перехода на новую строку использовался тег **<p>**

Рис.5 Визуальное отличие действия тегов **<p>** и **
**

<hr> - используется для создания на странице горизонтальной линии.

Теги, управляющие положением текста

<center>..</center> - расположение контента по центру.

<left>..</left> - расположение контента по левому краю.

<right>..</right> - расположение контента по правому краю.

Вопрос

Подумайте, когда лучше для выравнивания использовать теги, а когда атрибут **align** тега **<p>**?

Теги, управляющие форматированием текста

<I>..</I> - курсив.

.. - полужирный.

<U>..</U> - подчеркнутый.

<S>..</S> - зачеркнутый.
_{..} - нижний индекс.
^{..} - верхний индекс.

Для вставки символов, отсутствующих на клавиатуре, используются т.н. CER-последовательности, начинающиеся со знака &.

Например, § - §, ± - ± и т.д.

Полный список CER-последовательностей можно найти в интернете, или в папке «Дополнительные материалы».

Задание 1

Создайте html-страницу, соответствующую образцу «Образец.jpg» из папки «Задание2». Внимательно ознакомьтесь с пояснениям к заданию – файл «Задание.doc».

Задание 2

Создайте html-страницу по образцу «Образец.jpg» из папки «Задание2а». Обратите внимание, что в тексте встречаются символы, отсутствующие на клавиатуре – не забудьте воспользоваться CER-последовательностями. Будьте внимательны при форматировании страницы, старайтесь, чтобы ваша страница как можно больше соответствовала образцу.

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 3 Тема “Создание списков”.

Цель работы: Научиться создавать списки нумерованные и маркированные.

Список – перечень однотипных элементов. Существует два типа списков – нумерованные и маркированные. В нумерованных списках каждый элемент обозначается изменяющимися значениями (числами, буквами и т.д.). В маркированных списках все элементы обозначены одинаковыми символами (тире, галочки, точки и т.д.).

Маркированный список:

Маркированный список заключается в теги ... Тег означает начало нового элемента списка. Этот тег можно не закрывать.

```
<UL>
  <LI> первый элемент
  <LI> второй элемент
  <LI> третий элемент
  ...
  <LI> последний элемент
</UL>
```

Тег имеет атрибут **type**, позволяющий выбрать тип маркера. Возможны значения:

- disk - ● (по умолчанию);
- circle - ○;
- square - ■.

Пример

```
<html>
<head>
  <title>
    Пример списка
  </title>
</head>
<body>
  <UL>
    <LI> первый элемент
    <LI> второй элемент
    <LI> третий элемент
    <LI> последний элемент
  </UL>
</body>
</html>
```

Нумерованный список:

Нумерованный список заключается в теги **..**. Тег **** означает начало нового элемента списка. Этот тег можно не закрывать.

```
<OL>
  <LI> первый элемент
  <LI> второй элемент
  <LI> третий элемент
  ...
  <LI> последний элемент
</OL>
```

Тег **** имеет атрибут **type**, позволяющий выбрать тип нумерации. Возможны значения:

- 1 (по умолчанию);
- a;
- A;
- I;
- i.

Пример

```
<html>
<head>
  <title>
    Пример списка
  </title>
</head>
<body>
  <OL>
    <LI> первый элемент
    <LI> второй элемент
    <LI> третий элемент
    <LI> последний элемент
  </OL>
```

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

Задание 1

Создайте html-страницу, соответствующую образцу «Образец.jpg» из папки «Задание3». Будьте внимательны при форматировании текста.

Задание 2

Создайте html-страницу, соответствующую образцу «Образец.jpg» из папки «Задание3а». В качестве фона страницы должно быть изображение «main_bg.jpg». Заметьте, что текст страницы можно скопировать из файла «Текст.doc»

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа №4 Тема " Гиперссылки".

Цель работы: Научиться создавать гиперссылки.

Гиперссылка – это ссылка на другой документ или место в текущем документе. Ссылка на место в текущем документе называется анкером (якорем). Якорь используется обычно для организации меню на страницах, размер которых больше одного экрана (при нажатии на гиперссылку-якорь переход происходит на другую часть этой же страницы).

Создание обычной гиперссылки производится тегом **<A>** с атрибутом **href**, который указывает адрес места назначения. Текст, записанный между **<a>** и **** является собственно гиперссылкой.

Помимо обязательного атрибута href для тега **<a>** можно указывать и другие атрибуты, например **target** – параметр, который указывает, в каком окне откроется гиперссылка – текущем (**parent**) или новом (**blank**).

Пример

```
<html>
<head>
<title>
  Пример гиперссылки
</title>
</head>
<body>
<a href=agpu.net target=blank> Перейти на сайт АГПА </a>
</body>
</html>
```

Создание якоря производится тегом **<a>** с атрибутом **name=имя**. Ссылка на якорь осуществляется тегом **<A>** с атрибутом href, но перед именем якоря указывается знак #.

Пример

```
<html>
<head>
<title>
  Пример гиперссылки-якоря
</title>
</head>
<body>
<a href=#a1> Перейти к главе 1 </a> ссылка на якорь
...
<a name=a1> Глава 1 </a> сделали сам якорь
</body>
</html>
```

Задание 1

Создайте четыре html-страницы (папка Задание 4), связанные гиперссылками по следующей схеме: страница Index.html содержит гиперссылки на страницы I1.html, I2.html, I3.html. С каждой из этих страниц есть гиперссылка на страницу Index.html.

Задание 2

Создайте html-страницу, соответствующую образцу «Образец.doc» из папки «Задание4а». Страница содержит меню, гиперссылки в котором ссылаются на якоря, расположенные ниже в этом же документе.

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 5 Тема "Таблицы."

Цель работы: Научиться создавать и оформлять таблицы.

Тег `<table>..</table>` позволяет добавить таблицу на страницу. У этого тега определены атрибуты:

Align – размещение таблицы на странице. Возможны значения - **Left, right, center**.

Bgcolor – цвет заливки.

Border – толщина границы в пикселах. Если атрибут border не указан, то границ видно не будет.

Bordercolor – цвет границы.

Тег `<table></table>` указывает начало и окончание таблицы. Однако, если на странице прописать этот тег, таблица на самом деле создана не будет. Дело в том, что таблица создается только после создания строк таблицы и ячеек в них.

`<tr>..</tr>` - новая строка в таблице.

`<td>..</td>` - новая ячейка в строке.

Существуют несколько атрибутов, которые могут быть применены как к строке полностью, так и к отдельной ячейке:

Align – выравнивание содержимого внутри строки/ячейки по горизонтали. Возможны значения **Left, right, center**.

Valign - выравнивание содержимого внутри строки/ячейки по вертикали. Возможны значения **top** (по верхнему краю), **middle** (по центру), **bottom** (по нижнему краю).

Width – ширина строки/ячейки в пикселах или процентах. Если указаны проценты, ширина строки рассчитывается относительно размера таблицы, ширина ячейки рассчитывается относительно размера строки.

Height – высота строки/ячейки в пикселах или процентах.

Bgcolor – цвет заливки.

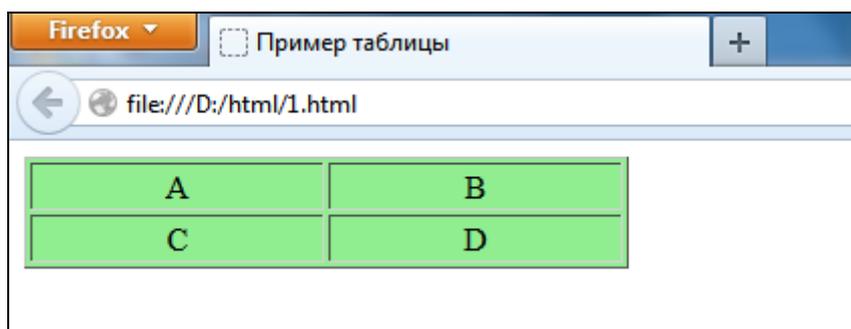
Bordercolor – цвет границы.

Background – фоновая графика.

Пример

```
<html>
<head>
<title>
Пример таблицы
</title>
</head>
<body>
<table border=1 bgcolor=lightgreen width=300>
<tr align=center>
<td> A </td>
<td> B </td>
</tr>
<tr align=center>
<td> C </td>
<td> D </td>
</tr>
</table>
</body>
</html>
```

В результате страница примет вид:



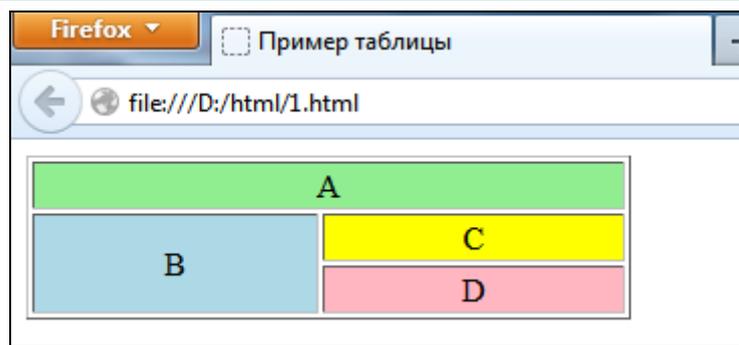
Часто возникает необходимость объединить некоторые ячейки. Это можно сделать с помощью атрибутов тега <td>:

Colspan – размах ячейки по горизонтали.

Rowspan – размах ячейки по вертикали.

Пример

Разработаем страницу следующего вида:



```
<html>
<head>
  <title>
    Пример таблицы
  </title>
</head>
<body>
  <table border=1 width=300>
    <tr align=center>
      <td colspan=2 bgcolor=lightgreen> A </td>
    </tr>
    <tr align=center>
      <td rowspan=2 bgcolor=lightblue> B </td>
      <td bgcolor=yellow> C </td>
    </tr>
    <tr align=center>
      <td bgcolor=lightpink> D </td>
    </tr>
  </table>
</body>
</html>
```

Задание 1

Создайте три любые (на выбор) html-страницы по образцу (папка Задание 5).

Задание 2*

Создайте четыре html-страницы, связанные гиперссылками, по образцу (папка Задание 5а). Заметьте, здесь таблица применяется не только для оформления меню, но и для разметки страницы в целом. В файле «Задание.doc» можно, выделив текст, посмотреть, какие параметры шрифта были использованы (тип шрифта, размер и т.д.)

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 6 Тема "Графика и мультимедиа".

Цель работы: Изучение правил добавления и размещения изображений.

.. - вставка изображения, видео или звука.

Атрибуты:

Src – имя файла

Width – ширина изображения в пикселах или процентах

Height – высота изображения в пикселах или процентах

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 7

Тема "Каскадные стили CSS".

Цель работы: Изучение синтаксиса стилей CSS.

CSS – Cascading Style Sheets (каскадные таблицы стилей) – язык стилей, определяющий отображение html-документа.

HTML используется для структурирования содержимого страницы. CSS используется для форматирования этого структурированного содержимого.

Базовый синтаксис CSS

Красный цвет абзаца с помощью html:

```
<p> <font color=red>...
```

Тот же результат с помощью CSS:

```
P{color:red}
```

Общий вид синтаксиса в CSS:

```
Selector {property: value}
```

Селектор (к какому тегу применяется свойство) Свойство значение

Применение CSS к html-документу

Существует 3 способа применить правила CSS к html-документу:

- 1) Инлайн (in-line) (атрибут style)

```
...  
<body>  
<p style="color:red">  
Текст абзаца  
</p>  
...
```

- 2) Внутренний (тег style)

```
<html>  
  <head>  
    <title>  
      Пример  
    </title>  
    <style type="text/css">  
      P {color:red}  
    </style>  
  </head>  
<body>  
  <p> текст абзаца</p>  
</body>  
</html>
```

- 3) Внешний (ссылка на таблицу стилей)

```
<html>
  <head>
    <title>
      Пример
    </title>
    <link rel="stylesheet" type="text/css" href="style.css">
  </head>
  <body>
    <p> текст абзаца</p>
  </body>
</html>
```

Документ style.css создается в Блокноте, имеет структуру:

```
P:{color:red;}
```

Шрифты

Font-family

указывает приоритетный список шрифтов, используемых для отображения текста. Если первый шрифт списка не установлен на компьютере, ищется следующий.

Например,

```
H1 {font-family: arial, verdana, sans-serif;}
```

```
H2 {font-family: "Times New Roman", serif;}
```

Font-style

Определяет стиль текста. Возможны значения:

- normal;

- italic;

Например,

```
H1 {font-family: arial, verdana, sans-serif; font-style:italic }
```

или

```
H1 {font: arial, verdana, sans-serif italic }
```

Font-variant

Определяет вариант шрифта:

- normal;

- small-caps (использует малые заглавные буквы вместо строчных)

Например,

```
H1 {font-variant: small-caps;}
```

Font-weight

Определяет вес шрифта:

- normal;

- bold;

- число от 100 до 900.

Например,

```
Td {font-weight:900;}
```

Font-size

Размер шрифта

Например,

```
H1 {font-size:30 px}-абсолютная единица
```

```
H2 {font-size:30 pt}-абсолютная единица
```

```
H3 {font-size:120%}-относительная единица
```

```
p {font-size:1 em}- относительная единица
```

Вместо

```
P{
  Font-style: italic;
  Font-weight: bold;
  Font-size: 30px;
  Font-family: arial,sant-serif;
}
```

Можно записать:

```
P{ Fon: italic bold 30px arial, sant-serif;}
```

Текст

Text-indent

Позволяет выделить параграф с помощью установки отступа для его первой строки.

Например,

```
P{text-indent:30 px;}
```

Text-align

Выравнивание текста. Возможны значения:

- left;
- right;
- center;
- justify.

Например,

```
H1{text-align:center}
```

Text-decoration

Добавление тексту декоративных эффектов

- underline;

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 8

Тема "Создание таблицы истинности с помощью JavaScript".

Цель работы: Научиться использовать основные арифметические функции

JavaScript для создания таблицы истинности операндов AND, OR, ->, ~.

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 9

Тема "Создание теста с помощью JavaScript".

Цель работы: Научиться создавать тесты, используя окна, операции ввода и вывода.

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа № 10

Тема "Создание простейших WEB-приложений средствами языка PHP".

Цель работы: Формирование навыков создания простейших web-приложений средствами языка PHP.

1. Создайте веб-приложение, реализующее калькулятор (минимум четыре действия).
Рекомендации: Создайте HTML-документ с формой, содержащей поля ввода для операндов и выпадающий список для выбора действия над операндами, а также кнопку, отсылающую информацию на сервер (не забудьте дать имена элементам формы). После этого создайте HTML-Создание простейших web-приложений средствами языка PHP документ с PHP-скриптом, который будет обрабатывать запрос из формы и выдавать результат. Не забудьте учесть ноль при делении!

2. Создайте веб-приложение, анализирующее введенное в поле ввода число по следующим параметрам: четность, знак (положительное\отрицательное) и количество разрядов. Если введено не число, то сообщить об этом.

Рекомендации: Создайте HTML-документ с формой, содержащей поле ввода для ввода анализируемого числа и кнопку, отсылающую информацию на сервер (не забудьте дать имена элементам формы). После этого создайте HTML-документ с PHP-скриптом, который будет обрабатывать запрос из формы и выдавать результат. Не забудьте про ноль! Помня о том, что PHP не следит за типом данных, число разрядов можно считать как количество символов в строке (если, конечно, пользователь ввёл число).

3. Создайте веб-приложение, позволяющее проверить информацию, введенную в регистрационную форму, состоящую из следующих полей: Имя (максимум 10 символов) и адрес электронной почты (наличие обязательного символа @ и минимум двух точек). Если хотя бы одно поле содержит ошибку, то вывести соответствующее сообщение, если же вся информация введена верно, то поздравить с успешной регистрацией.

Рекомендации: Создайте HTML-документ с формой, содержащей нужное количество полей ввода и кнопку, отсылающую информацию на сервер (не забудьте дать имена элементам формы). После этого создайте HTML-документ с PHP-скриптом, который будет обрабатывать запрос из формы и выдавать результат.

4. Создайте веб-приложение, реализующее тест на основе выпадающих списков.

Рекомендации: Создайте HTML-документ с формой, содержащей нужное количество выпадающих списков с вариантами ответов на вопросы и кнопку, отсылающую информацию на сервер (не забудьте дать имена элементам формы). После этого создайте HTML-документ с PHP-скриптом, который будет обрабатывать запрос из формы и выдавать результат.

Форма представления отчета:

Студент должен создать проект, продемонстрировать работу и ответить на контрольные вопросы.

Лабораторная работа №11

Контрольные вопросы

- 1) Типы данных.
- 2) Переменные.
- 3) Типы данных.
- 4) Константы.
- 5) Комментарии.
- 6) Операторы.

Типовые задания

1. Написать скрипт, который выполнял бы элементарные арифметические действия (сложение, умножение, вычитание, деление) и вывод результата на экран.

Решение:

```
<form method="post" name="form1">
```

Действие:<input type="radio" name="arifm" value="1" checked> +

```
<input type="radio" name="arifm" value="2"> -
```

```
<input type="radio" name="arifm" value="3"> *
```

```

<input type="radio" name="arifm" value="4"> / <br>
Введите a:<input type="text" name="a"><br>
Введите b:<input type="text" name="b"><br>
<input type="submit" name="button" value="Расчет">
</form>
<?
if (isset($_POST['button']))
{
switch($_POST['number'])
{
case 1:
{
$result=$_POST['a']+$_POST['b'];
break;
}
case 2:
{
$result=$_POST['a']-$_POST['b'];
break;
}
case 3:
{
$result=$_POST['a']*$_POST['b'];
break;
}
case 4:
{
if ($_POST['b']!=0) {$result=$_POST['a']/$_POST['b'];}
else {$result="Знаменатель не должен равняться нулю!";}
break;
}
}
}
echo 'Результат: '.$result;
}
?>

```

Задачи для решения

1. Организовать проверку двух введенных чисел. Если а меньше b, то вывести число а меньше b, и если больше, то соответственно. Пример результата ”Число 4 меньше 7”
2. Написать скрипт вычисления корней квадратного уравнения

Лабораторная работа №12

Контрольные вопросы

- 1) Строковый тип данных.
- 2) Основные операции со строками
- 3) Функции поиска в тексте.
- 4) Функции форматного вывода.
- 5) Функции работы с URL.
- 6) Функции преобразования регистра.

Типовые задания

Найти количество вхождений фрагмента в строку.

Решение.

```
<?
$str = "dfhd@ffs@dfskfk@asas";
$substr_count = substr_count($str,"@");
echo ($substr_count);
?>
```

Задачи для решения

1. Организовать поиск. Слова должны обрезаться на 2 символа, чтобы искалось к примеру не «Тюменский», а «Тюменск», т.к. в тексте может встречать Тюменскую область, Тюменский район.

Лабораторная работа №13

Контрольные вопросы

- 1) Понятие массив.
- 2) Способы объявления массива.
- 3) Обход элементов массив.
- 4) Реализация многомерного массива.
- 5) Поиск значения в массиве.

Типовые задания

1. Обойти все элементы массива и вывести их на экран.

Решение.

```
<?
$array = array("Мышь", "Клавиатура", "Монитор","ИБП");
foreach($array as $index => $val)
{
    echo("$index -> $val <br>");
}
?>
```

Задачи для решения

1. Организовать поиск заданного значения в массиве. Вывести весь массив, причем искомый элемент должен быть выделен жирным.

Лабораторная работа №14

Контрольные вопросы

- 1) Понятие функции.
- 2) Входные параметры функции.
- 3) Вызов функции.
- 4) Рекурсия.

Задачи для решения

1. Написать функцию вычисления корней квадратного уравнения. Соответственно с входными параметрами **a, b, c**.

Лабораторная работа №15

Работы с файлами в php

Функции работы с фалами.

Контрольные вопросы

- 1) Открытие файлов.
- 2) Отображение файлов.

- 3) Заккрытие файлов.
- 4) Чтение из файлов и запись в файлы.
- 5) Копирование, переименование и удаление файлов.
- 6) Атрибуты файлов.

Типовые задания

1. Вывести содержимое файла на экран.

Решение.

```
<?
$file = fopen("c:/www/html/pavlovo.jpg","rb");
if(!file)
{ echo("Ошибка открытия файла"); }
else
{ fpassthru($file); }
?>
```

2. Записать в файл текстовую строку.

Решение.

```
<?
$file = fopen ("file.txt","r+");
$str = "Hello, world!";
if ( !$file )
{ echo("Ошибка открытия файла");}
else
{ fputs ( $file, $str); }
fclose ($file);
?>
```

Задачи для решения

1. Написать скрипт ведения адресной книги, где хранится ФИО человека, номер телефона и его адрес. Все данные хранить в текстовом файле по шаблону:

Иванов И. И||33-33-33||Моторостроителей 33 кв .4

Петров П. П.||35-35-35||Энергетиков 123 кв .77

и т.д.

Лабораторная работа №16

Типовые задания

1. Вывести содержимое таблицы clients с сортировкой по зарплате по убыванию, при условии, что зарплата выше 10 т.р.

Решение.

```
<?
$result=mysql_query("SELECT * FROM `clients` WHERE `zp`>'10000' ORDER BY `zp`
DESC ");
$count=mysql_num_rows($result);
$value=mysql_fetch_array($result);
for ($i=1; $i<=$count; $i++)
{
echo $value['fio'].' | '.$value['zp'].'<br>';
$value=mysql_fetch_array($result);
}
mysql_free_result($result);
```

?>

Задачи для решения

1. Написать скрипт новостей. Сортировка новостей по дате. Вывод последних пяти новостей (SELECT.....LIMIT 5), Добавление новости, редактирование новости, удаление новости через web интерфейс. Поиск новости.