

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Пономарева Светлана Викторовна
Должность: Проректор по УР и НО
Дата подписания: 20.09.2023 20:50:37
Уникальный программный ключ:
bb52f959411e64617366ef2977b97e87139b1a2d



**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»
(ДГТУ)
АВИАЦИОННЫЙ КОЛЛЕДЖ**

УТВЕРЖДАЮ

Директор колледжа

А.И. Азарова

личная подпись

инициалы, фамилия

«__» _____ 2021г.

**Методические рекомендации по выполнению
практических работ по дисциплине**

ОП.17 Веб-программирование

для студентов специальности

09.02.07 Информационные системы и программирование

Ростов-на-Дону
2021 г.

СОДЕРЖАНИЕ

1.	Изучение основ разработки HTML- документа	4
2.	Формирование гипертекстовых и графических ссылок	5
3.	Работа с таблицами	6
4.	Создание фреймов. Формирование Web-страниц	7
5.	Объект IMAGE в JavaScript	8
6.	Создание динамических Web-страниц	10
7.	Установка Apache и PHP	11
8.	Работа с функциями	12
9.	Работа с массивами	13
10.	Работа с формами	14

Практическое занятие № 1

Тема: Изучение основ разработки HTML- документа

Цель: уметь осуществлять физическую разметку текстов, формирование таблиц.

Теоретическая часть

HTML (Hypertext Markup Language, язык разметки гипертекста) – язык компоновки документов и спецификации гиперссылок, используемый для кодирования документов в системе World Wide Web. Средствами HTML задаются синтаксис и размещение специально встроенных указаний, в соответствии с которыми браузер отображает содержимое документа: текст, изображения и данные других типов, поддерживаемых данным браузером. Текст самих встроенных указателей WEB-браузером не отображается. В языке HTML реализована поддержка механизма гипертекстовых ссылок, которые обеспечивают связь данного документа с другими документами (они могут находиться в локальной системе, в системе World Wide Web или могут быть получены с помощью других ресурсов Internet – FTP, Gopher и т. д.).

HTML-документ состоит из текста, который является содержимым создаваемого документа, и тегов, определяющих структуру и внешний вид документа при его отображении браузером.

Практическая часть

1. Создайте структуру HTML – документа, используйте указанные теги:

```
<HTML>
<HEAD> <title> </title> </HEAD>
<body>
P
U
BIG
Small
Sub
Sup
Strike
TT
MARQUEE
NOBR</body>
</HTML>
```

2. В тело созданного вами документа вставьте элементы формирования списков:
нумерованных;
нenumерованных;
списков определений.

3. Сформируйте в теле документа таблицу, используйте элементы:
TABLE с атрибутами align, background, bgcolor, border, bordercolor, cellpadding, cellspacing, rules
TR
TH с атрибутами colspan, rowspan.
TD
CAPTION

Анализ полученных результатов

Сравнить результаты работы использованных вами тегов с различными значениями атрибутов. Сделайте разметку таблицы невидимой и убедитесь, что этот режим удобен для создания элементов web-страницы, обеспечивая для разных систем отображения одинаковый вид ее структуры.

Контрольные вопросы

1. Что такое тег META и зачем он нужен?
2. Что такое тег BASE и зачем он нужен ?
3. Какой тег используется для объединения нескольких столбцов в таблице?
4. Каким образом можно внутри ячейки таблицы уменьшить расстояние от текста до ее границы?

Практическое занятие № 2

Тема: Формирование гипертекстовых и графических ссылок

Цель: уметь создавать гипертекстовые и графические ссылки в HTML-документах.

Теоретическая часть

Гиперссылки указывают браузеру на адрес документа, который должен быть загружен в браузер, после активизации ссылки нажатием кнопкой мыши или клавиши ввода с клавиатуры. Гиперссылки бывают текстовые и графические.

Для указания текстовых ссылок в языке HTML используется элемент A (anchor — якорь). Важнейшим атрибутом данного элемента является href, указывающий адрес ресурса, на который производится ссылка. Закрывающий тег обязателен.

```
<BODY>
```

```
<P>
```

```
Здесь находится <A href="http://www.bhv.ru">сайт Издательства BHV </A>
```

```
</P>
```

```
</BODY>
```

Иногда бывает очень полезным определять ссылки на некоторые разделы страниц.

Практическая часть

1. В документ, созданный на предыдущем занятии вставьте гиперссылку

`` на C2.html файл `` , `` на C3.html файл `` , `` на C4.html файл ``. Создайте эти документы и разместите в соответствии со ссылками на них.

2. Создайте ссылку на конкретное место текущего HTML-документа. Место на которое организуете ссылку выберите за пределами текущей страницы.

3. Создайте ссылку на конкретное место другого HTML-документа. Место на которое организуете ссылку выберите за пределами первой страницы.

4. С помощью тега `` вставьте картинку в ваш документ.

5. Измените содержимое тега IMG таким образом:

```
<IMG SRC="имя_файла. jpeg " USEMAP="#map_name">
```

USEMAP описывает где находится карта описываемого изображения (в нашем случае она находится в данном HTML, если бы была в другом месте, то синтаксис был бы таким:
...USEMAP= "URI #map_name">

6. Введите описание карты в HTML- документ.

```
<MAP NAME="map_name">
```

```
<AREA Shape="RECT" coords="0,0,50%,50%" href="имя файла.html">
```

```
</MAP>
```

«имя файла.html» – имя файла на который организуется ссылка. Создайте этот файл. Содержимое его будет пояснением той части рисунка, которую Вы связали с этим файлом.

Атрибут Shape – указывает форму определяемой области изображения. Может принимать значения:

Rect – прямоугольник

Circle – круг

Poly – многоугольник произвольной формы

Атрибут Coords – указывает координаты в пикселях (или в % от всего рисунка). Для прямоугольника это четыре координаты верхнего левого и правого нижнего углов. Для круга – три координаты (две – центр круга, третья – радиус). Для многоугольника - это описание каждого угла в координатах. Координаты считаются с нуля. Создайте несколько областей с различными значениями Shape.

Атрибут href="URL" – (в нашем случае это имя файла) описывает ссылку, действия по которой будут выполняться при щелчке в заданной области.

7. Сформируйте несколько областей на рисунке разной формы (меняя атрибут Shape) и каждую из них свяжите со своим HTML-документом.

Анализ полученных результатов

Сравните результаты, полученные в ходе выполнения задания и прокомментируйте в каких ситуациях возможно использование каждого из изученных механизмов.

Контрольные вопросы

1. Что такое гипертекстовая ссылка?
2. Что такое графическая ссылка?
3. Как еще можно использовать изображение в качестве ссылки?

Практическое занятие № 3

Тема: Работа с таблицами

Цель: получить теоретические знания и практические навыки в использовании технологии HTML и таблиц.

Теоретическая часть

Парный тег <TABLE> и </TABLE> используется для описания таблиц. Ячейки таблиц в языке HTML могут содержать любые HTML-элементы: заголовки, списки, текстовые абзацы, графику и элементы форм.

Тег <TABLE> имеет атрибуты:

- align="left|center|right" – определяет положение подписи таблицы;
- bgcolor="Цвет" – задает фоновый цвет;
- border="Толщина границы в пикселах" – определяет вид границы таблицы;
- cellpadding="Расстояние между содержимым ячейки и ее краев" – определяет расстояние от границ ячейки до ее содержимого в пикселах;
- cellspacing="Расстояние между ячейками таблицы" – определяет расстояние между ячейками в пикселах;
- width="Ширина" – определяет ширину таблицы.

Для создания строк и столбцов таблицы служат парные теги <TR> и </TR>, <COL> и </COL>. Для создания ячейки таблицы служит парный тег <TD> и </TD>. Для создания заголовка таблицы служит парный тег <TH> и </TH>. Для группирования элементов служат теги <COLGROUP>, <TBODY> и <TFOOT>.

Практическая часть

1. Продумайте структуру сайта «Модельный ряд». Выполните поиск графического и текстового материала для информационного контента сайта.
2. Создайте html – документ с использованием атрибутов таблицы.
3. Введите описание стилей body (фоновая картинка), h2 (тип шрифта в ячейке таблицы), .text (заголовок), .t1 (стиль рамки для первой ячейки), .t2 (стиль рамки для второй ячейки), .link (стиль для шрифта текста, который является гиперссылкой), .d1 (абсолютное положение таблицы на странице), a:visited и a:hover (цвет гиперссылок).
4. Создайте web-страницы. Разместите на страницах графические изображения.

Анализ полученных результатов

Проанализируйте результаты работы и подумайте какие дополнительные функции можно реализовать в рамках предложенных заданий.

Контрольные вопросы

1. Как объединить несколько столбцов таблицы?
2. Как объединить несколько строк таблицы?
3. Как выравнивать текст внутри ячейки?
4. Как вставить картинку на фон таблицы?

Практическое занятие № 4

Тема: Создание фреймов. Формирование Web-страниц

Цель: получить теоретические знания и практические навыки в использовании технологии HTML и фреймов.

Теоретическая часть

Фрейм – элемент языка HTML, позволяющий разделить web-страницу на несколько независимых окон, в каждом из которых размещают отдельную web-страницу. Для создания фреймов используются дескрипторы `<FRAMESET>` и `<FRAME>`.

Тег `<FRAMESET>` описывает все фреймы, на которые делится экран. Экран можно разделить на несколько вертикальных или несколько горизонтальных фреймов. Для описания каждого фрейма в отдельности служит тег `<FRAME>`.

Тег `<FRAMESET>` имеет атрибуты:

- `cols` – задает количество и размеры столбцов в создаваемом наборе фреймов;
- `rows` – задает количество и размеры строк в создаваемом наборе фреймов в пикселах, в процентах или связанных масштабных коэффициентах;
- `border` – задает толщину обрамления в пикселах для всех фреймов, входящих в данный набор фреймов;
- `frameborder` – позволяет отменить вывод обрамления у фреймов, независимо от значения атрибута `border`.

Создание веб-страницы, содержащей два горизонтальных фрейма.

```
<HTML><HEAD><TITLE> Два горизонтальных фрейма </TITLE>
</HEAD> <FRAMESET FRAMEBORDER=1 BORDER="5"
ROWS="480, 200"><FRAME SRC="1.htm">
<FRAME SRC="2.htm"></FRAMESET></HTML>
```

Практическая часть

1. Продумайте структуру сайта “ЭЛЕКТРОННЫЙ тест проверки знаний студентов изучаемых дисциплин”. Выполните поиск графического и текстового материала для информационного контента сайта. Главная страница `Index.html` должна состоять из двух фреймов, в первом из которых размещается меню (список вопросов), а во втором – изначально выводится иллюстрация по теме тестирования, а при выборе вопросов в меню выводится текст вопросов.
2. Создайте страницу с именем `1.html`, соответствующую первому вопросу. Она должна состоять из двух фреймов, в которые загружаются страницы с именами `1-1.html`, `2-1.html`.
3. Аналогичным способом создайте страницы, соответствующие остальным вопросам. Они также должны состоять из двух фреймов, в которые загружаются страницы с соответствующими именами.
4. Для автоматизации подсчета правильных ответов на вопросы ограниченные возможности HTML дополняют сценариями. Сценарий (скрипт) может быть написан на языке JavaScript или VBScript и представляет собой программный код для создания динамических web-страниц. Преимущество сценариев – простота. Чтобы добавить код сценария в HTML-документ, используют тег `<script>`, который располагается в пределах тегов `<head></head>` или `<body></body>`.
5. После последнего вопроса необходимо создать две кнопки: кнопку «Проверить» (при нажатии которой в текстовой строке Результат выводится количество правильных ответов) и кнопку сброса «Стереть», нажатие которой отменяет все сделанные изменения.

Анализ полученных результатов

Сравните результаты, полученные в ходе выполнения задания и прокомментируйте в каких ситуациях возможно использование каждого из изученных механизмов.

Контрольные вопросы

1. Какова структура тэга `<FRAMESET>` в HTML-документе? Что он может содержать?
2. С помощью каких тэгов можно создавать маркированные и нумерованные списки?
3. Как заставить в браузере отображать различные служебные символы HTML-страницы (например, угловые скобки, кавычки, апострофы)?

Практическое занятие № 5

Тема: Объект IMAGE в JavaScript

Цель: уметь создавать навигационные изображения при разработке сценариев смены изображений в HTML-документах с использованием Jscript.

Теоретическая часть

При создании Web- страниц широко используются сменяющиеся изображения. Установка курсора на изображение приводит к замене на "подсвеченную" его версию. Когда курсор убирается, то возвращается первоначальная версия изображения. Такой прием используется в навигационных панелях.

Практическая часть

1. Создание сменяющихся изображений происходит в три этапа:

создание объектов IMAGE;

построение функций смены изображения;

формирование обработчиков событий.

Создание объектов IMAGE

Предварительно создайте в любом графическом редакторе изображения трех отделений РЭТК: ПВМ, АТП, МЭП. Они будут использоваться для вывода информации об этих отделениях. В качестве таких изображений могут использоваться графические кнопки либо пиктограммы, отображающие специализацию отделений. Имена этих файлов должны быть: pvm_on.gif, atp_on.gif и mp_on.gif соответственно. Эти изображения будут появляться при наведении курсора на первоначальные "неподсвеченные" такие же графические объекты, только имеющие иное цветовое решение. Эти первоначальные графические объекты также создайте с помощью графического редактора. Цвет подберите таким, чтобы он сочетался с выбранным цветом страницы, но не сливался с ней и был хорошо виден. Эти файлы должны иметь имена: pvm_off.gif, atp_off.gif и mp_off.gif соответственно. Теперь переходите к созданию объектов IMAGE.

Создадим HTML- страницу, содержащую три навигационных изображения отделений РЭТК без обработчиков событий.

```
<BODY>
<A HREF="pvm/index.html"><IMG src="images/h_pvm_off.gif"
WIDTH="71" HEIGHT="33" BORDER="0" NAME="pvm"
ALT="отделение ПВМ"></A>
<A HREF="atp/index.html"><IMG src="images/h_atp_off.gif"
WIDTH="71" HEIGHT="33" BORDER="0" NAME="atp"
ALT="отделение АТП"></A>
<A HREF="mp/index.html"><IMG src="images/h_mp_off.gif"
WIDTH="71" HEIGHT="33" BORDER="0" NAME="mp"
ALT="отделение МЭП"></A>
```

Теперь построим объекты IMAGE для изображения pvm, предварительно проверив поддерживает ли браузер пользователя сменяющиеся изображения. Сценарий поместим в раздел HEAD.

```
<HEAD>
<SCRIPT language="JavaScript">
<!-- браузеры ранних версий игнорируют следующий код
//создание объектов IMAGE
if(document.images) /* проверяется поддерживает ли браузер объект IMAGE. Если поддерживает,
то будет возвращено значение истина, в обратном случае - ложь*/
{
    pvmOn=new Image (71,33); /* создали новый объект IMAGE с именем pvmOn при подсвеченном
состоянии*/
    pvmOn.src=" images/pvm_on.gif"; /* назначили реальное изображение для созданного объекта*/
    pvmOff=new Image (71,33); /* создали новый объект IMAGE с именем pvmOff при обычном
неподсвеченном состоянии*/
    pvmOff.src=" images/pvm_off.gif"; /* назначили реальное изображение для созданного объекта*/
    ...
}
//здесь сокрытие кода прекращается-->
</SCRIPT >
</HEAD>
```

Для смены изображения pvm сделано все. Сделайте аналогично и для остальных двух сменяющихся изображений.

Построение функций смены изображения

При наведении пользователем курсора на одно из изображений, являющееся пунктами меню, обработчик событий JavaScript должен вызвать функцию, которая осуществит замену на подсвеченное изображение (графика с обозначением On). Когда курсор уходит с изображения должно возвращаться «неподсвеченное»

изображение (графика с обозначением Off). Т.е. необходимо создать две функции, которые предназначены для всей страницы и потому должны располагаться в разделе HEAD сразу после объектов IMAGE.

```
function on(pic)
{
  if (document.images)
  {
    document.images[pic].src=eval(pic+"on.src");    /* метод eval применяется для соединения значения pic (оно передается в функцию и может быть равным pvm, atr_ или mp) со строкой on.src. В результате получится новое значение, присваиваемое в качестве источника изображения, над которым проходит курсор*/
  }
}
```

Как только пользователь уберет курсор, нужно вызвать другую функцию. Назовем ее off.

```
function off (pic)
{
  if (document.images)
  {
    document.images[pic].src=eval(pic+"off.src");    /* метод eval применяется для соединения значения pic (оно передается в функцию и может быть равным pvm, atr_ или mp) со строкой off.src. В результате получится новое значение, присваиваемое в качестве источника изображения, с которого убирается курсор*/
  }
}
```

Формирование обработчиков событий

События, которые мы хотим использовать на странице, заключаются в смене изображений при наведении мыши на изображение или удалении ее с изображения. Обработчики этих событий должны располагаться в теге якоря. Введем обработчик события onMouseOver в тег якоря отделения ПБМ:

```
<A HREF="pvm/index.html" onMouseOver="on('pvm'); window.status=' отделение ПБМ '; return true;"><IMG src="images/h_pvm_off.gif" WIDTH="71" HEIGHT="33" BORDER="0" NAME="pvm" ALT="отделение ПБМ"></A>
```

При наведении курсора на ссылку "отделение ПБМ" обработчик события вызовет функцию on() с передаваемым в функцию значением. Для отделения ПБМ это будет on('pvm'). Оператор window.status=' отделение ПБМ' в строке состояния браузера выведет соответствующую фразу. Оператор *return true* сообщает интерпретатору о завершении всех действий в обработчике. Теперь добавим обработчик, меняющий изображение когда курсор мыши удаляется с изображения: onMouseOut.

```
<A HREF="pvm/index.html" onMouseOver="on('pvm'); window.status=' отделение ПБМ '; return true;" onMouseOut="off('pvm'); window.status=' '; return true;"><IMG src="images/h_pvm_off.gif" WIDTH="71" HEIGHT="33" BORDER="0" NAME="pvm" ALT="отделение ПБМ"></A>
```

Для данного обработчика вызывается функция off(), ей передается имя графического изображения 'pvm'. Фраза из статус строки удаляется оператором window.status=' '. После чего происходит выход из обработчика с помощью оператора *return true*.

Произведите аналогичные добавления обработчиков событий для остальных отделений.

2. В представленном коде необходимо найти и устранить ошибку.

Анализ полученных результатов

Проанализировать результаты работы и определить тип найденной ошибки.

Контрольные вопросы

1. Что такое меняющиеся изображения?
2. В какой части HTML-документа расположен обработчик события?
3. Зачем используется метод eval()?

Практическое занятие № 6

Тема: Создание динамических Web-страниц

Цель: уметь создавать динамические Web-страницы с использованием JavaScript

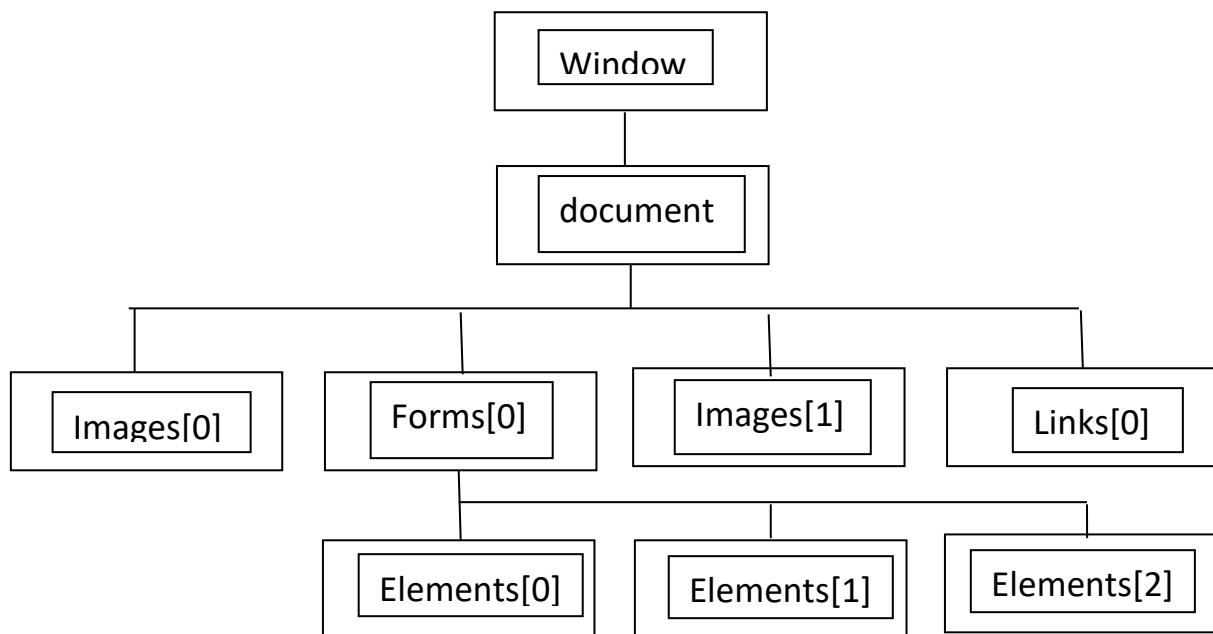
Теоретическая часть

При генерации страниц в Web возникает дилемма, связанная с архитектурой "клиент-сервер". Страницы можно генерировать как на стороне клиента, так и на стороне сервера. Последнее реализуется через механизм подстановок на стороне сервера. Генерация страниц на стороне клиента может быть реализована с помощью языка программирования JavaScript. Важность создания динамических страниц с его помощью сегодня вновь приобрела популярность, т.к. снижается число обращений к серверу при реализации функций, которые могут быть решены с помощью ресурсов клиента, и повышается скорость реакции на запросы пользователя.

JavaScript-язык управления сценариями просмотра гипертекстовых Web страниц на стороне клиента. JavaScript изменяет значения атрибутов HTML-тегов и свойств среды отображения при просмотре HTML-страницы пользователями. При этом не происходит перезагрузка страницы. На практике это выглядит как: -изменение цвета фона страницы; -открыть новое окно; -выдать предупреждение; -поместить картинку в документ. Для реализации этой идеи используется объектная модель документа. Суть модели в том, что каждый HTML-код-объект,

Доступ к объекту в JavaScript осуществляется в соответствии с положением объекта в иерархии созданного документа начиная со старшего:

(window.document.forms[0].elements[2]-обращение к кнопке)



Когда пользователь внесет содержимое в текстовые поля и нажмет на кнопку "Отправить", запустится сценарий JavaScript, расположенный в action формы. Поэтому сценарию будет выдано приветствие пользователю с именем, введенным им.

Практическая часть

1. Создайте два вертикальных фрейма.

2. В левый фрейм поместите файл, в котором опишите меню, содержащее завтрак, обед и ужин.

В правый поместите файл, который выводит текущую дату, время и приветствие: доброе утро, добрый день, добрый вечер, в зависимости от текущего времени суток. Утро с 6:00 до 12:00, день с 12:00 до 18:00, вечер с 18:00 до 1:00.

3. Выбор пункта меню в левом фрейме должен в правый фрейм загружать соответствующее меню.

4. Составьте отчет о проделанной работе. Он должен содержать листинги всей программ и описание структуры созданного вами программного обеспечения.

Анализ полученных результатов

Проанализируйте результаты работы и прокомментируйте их с точки зрения использования, например, для создания сайта для кафе или ресторанов.

Контрольные вопросы

1. Что такое объект?
2. Что позволяет делать иерархическая структура объектов JavaScript ?
3. Какой способ организации обращений к элементам документа на ваш взгляд лучше и чем?

Практическое занятие № 7

Тема: Установка Apache и PHP

Цель: уметь устанавливать сервер Apache и PHP

Теоретическая часть

Для установки Apache, PHP, MySQL, phpMyAdmin и других компонентов используемых Web-разработчиками (программистами и дизайнерами) для отладки сайтов на «домашней» (локальной) Windows-машине без необходимости выхода в Интернет был разработан *Джентльменский набор Web-разработчика* («Denwer», читается «Денвер»). Это бесплатный пакет, чем он еще более привлекателен и устанавливает сервер Apache поверх Windows.

Практическая часть

1. Изучить статью по установке базового комплекта Denwer-2, размещенную по адресу, указанному в кратких сведениях.
2. Ответьте на вопросы:
Для чего нужен Denwer?
Что содержит базовый комплект Denwer?
Как проверить готовность компьютера к работе с сетью?
Какие возможны проблемы и как их устранить?
Как осуществить установку дистрибутива?

Произведите установку дистрибутива в соответствии с инструкцией. Помните, что установку нужно провести только с запуском по явной команде старта (по щелчку на ярлыке запуска на рабочем столе) и отключать диск при остановке!!!

Как проверить работает ли комплекс?

Как добавить новый хост?

Откройте статью Denwer.htm, расположенную по адресу: :\\Install\\Apache\\2\\ ApacheРуководство\\1\\Основные возможности проекта «Denwer.htm», изучите ее и ответьте на вопросы:

- Что такое автономность Denwer(a) и в чем она заключается?
- Что такое виртуальный диск и зачем он нужен?
- Что создается в директориях- /home,etc,usr?
- Зачем нужен локальный сервер?
3. Создайте папку в Webservers\\home\\ , например t1.ru . В ней создайте папку WWW, в которую будете записывать файлы на PHP.
4. Файлы на PHP можно создавать в блокноте и записывать в хост t1.ru\\WWW\\NameFale.php.
5. Запуск PHP- файлов производите из IE, войдя в папку Webservers и набрав URL http://t1.ru/NameFale.php.
6. Создайте на полученное задание PHP-программу, устраните возможные ошибки.
7. Оформите отчет, поместив в него ответы на вопросы и листинг PHP программы созданной вами.

Анализ полученных результатов

Проанализируйте результаты работы и прокомментируйте их с точки зрения простоты и удобства использования пакета Denwer.

Контрольные вопросы

1. Что такое пакет Denwer?
2. Что такое phpMyAdmin и зачем он нужен?
3. Какие изменения произошли в структуре каталога вашей машины после установки пакета Denwer?

Практическое занятие № 8

Тема: Работа с функциями

Цель: уметь создавать и использовать функции в PHP

Теоретическая часть

Функция-это блок инструкций, который может быть вызван из программы. При вызове блок инструкций выполняется. Функции можно передать некоторые значения, она их обработает и может вернуть полученные значения в программу ее вызвавшую.

Например, часто бывает необходимо создавать панель навигации. Чтобы не формировать каждый раз код достаточно написать функцию, создающую эту панель и при необходимости вызывать ее.

Практическая часть

Создайте php – документ:

```
<html>
<head> <title>Работа с функцией</title> </head>
<body>
<?php
Echo "<H3>Добро пожаловать! </H3><br>"
Function navig()
```

```
{
{
$number=$number*10;
}
```

```
ten();
print $number;
```

1. Определите, что будет выведено на экран в результате выполнения следующего фрагмента кода?

```
$number=50;
Function ten()
```

```
{
Global $number;
$number=$number*10;
}
```

```
ten();
print $number;
```

2. Установите, что будет выведено на экран в результате выполнения следующего фрагмента кода?

```
$number=50;
Function ten($n)
```

```
{
$n=$n*10;
}
```

```
ten($number );
print $number;
```

3. Проверьте, что будет выведено на экран в результате выполнения следующего фрагмента кода?

```
$number=50;
Function ten(&$n)
```

```
{
$n=$n*10;
}
```

```
ten($number );
print $number;
```

4. Напишите функцию, которая принимает четыре текстовые строки, а возвращает строку HTML-текста, создающую таблицу, где каждая из исходных строк расположена в отдельной ячейке.

5. Получите индивидуальное задание на создание функции и создайте программу.

6. Представьте отчете комментарий по полученным результатам и листинг программ индивидуальных заданий.

Анализ полученных результатов

Проанализируйте результаты работы и прокомментируйте их с точки зрения возможностей, которые обеспечивают функции.

Контрольные вопросы

1. Что такое функция?
2. Что такое функция возвращающая значение?
3. Как осуществляется вызов функции?

Практическое занятие № 9

Тема: Работа с массивами

Цель: уметь создавать и использовать массивы в PHP

Теоретическая часть

Массив - набор значений, имеющих одно имя. Доступ к конкретному значению можно осуществить по его номеру. Массивы позволяют эффективно производить сортировки значений, их просмотр и поиск. Нумерация элементов в массиве начинается с нуля. Определение массива осуществляется функцией `array()`. Массивы, к элементам которых можно обратиться по имени называются ассоциированными. Массив, каждый элемент которого является массивом называется многомерным. Размер массива можно определить функцией `count()`. Просмотреть простой массив можно с помощью инструкции `foreach ($имя -массива as $имя -переменной)`. Просмотр ассоциированного массива осуществляется инструкцией `foreach ($имя-массива as $key=>$value)`. Объединение массивов производится функцией `array_merge()`, а добавление элементов с помощью функции `array_push()`. выделить часть массива можно функцией `array_slice()`. Сортировка простого массива производится функцией `sort()`, ассоциированного функцией `asort()`. Для сортировки ассоциированных массивов по именам полей используется функция `krsort()`.

Практическая часть

1. Создайте многомерный массив из Ф.И.О студентов вашей группы, отсортируйте в алфавитном порядке. Сделайте массив ассоциированным, сопоставив каждому студенту оценки по предметам, выставяемые ежемесячно.
2. Организуйте вывод на экран списка отличников, хорошистов, неуспевающих за конкретный месяц.
3. Создайте многомерный массив, содержащий название фильмов, организованных по жанрам. Пусть этот массив будет ассоциированным, имена полей в нем будут разными жанрами («мелодрама», «боевик», «детектив» и пр.), а элементами - названия фильмов, фамилия режиссера, год выпуска.
4. Организуйте вывод на экран массива отсортированного по Ф.И.О режиссера, по жанрам и в порядке их выхода.
5. Создать на сервере массив вопросов. Каждый вопрос имеет по четыре варианта ответов, один из которых верный. Студент регистрируется (вводит номер группы и ФИО). С сервера пересылается 3 вопроса случайным образом выбранных из массива вопросов. Студент выбирает верные варианты ответов и отправляет на сервер. Там происходит проверка и студенту отправляется информация о том, сколько было верных ответов.
6. Результаты оформите в отчет. Представьте в отчете комментарий по полученным результатам и листинг программ.

Анализ полученных результатов

Проанализируйте результаты работы и подумайте какие дополнительные функции можно реализовать в рамках предложенных заданий.

Контрольные вопросы

1. Что такое ассоциированный массив?
2. Что такое многомерный ассоциированный массив?
3. Как осуществляется сортировка ассоциированного массива по именам?

Практическое занятие № 10

Тема: Работа с формами

Цель: уметь создавать и использовать формы для организации клиент-серверного взаимодействия

Теоретическая часть

Тег <FORM> устанавливает форму на веб - странице. Форма предназначена для обмена данными между пользователем и сервером. Область применения форм не ограничена отправкой данных на сервер, с помощью клиентских скриптов можно получить доступ к любому элементу формы, изменять его и применять по своему усмотрению.

Документ может содержать любое количество форм, но одновременно на сервер может быть отправлена только одна форма. По этой причине данные форм должны быть независимы друг от друга.

Для отправки формы на сервер используется кнопка Submit. Когда форма отправляется на сервер, управление данными передается CGI-программе, заданной параметром action тега <FORM>. Предварительно браузер подготавливает информацию в виде пары «имя=значение», где имя определяется

Практическая часть

1. Создайте html – документ для отправки на сервер имени, возраста и e-mail-а пользователя в программу n2.php, используя метод get.
2. Программа n2.php должна отправить приветствие пользователя с использованием его имени и возраста
3. Прокомментируйте полученный результат.
4. В форме измените метод отправки на сервер данных.
5. Изменился ли результат работы файла n2.php?
6. Прокомментируйте.
7. Измените предыдущую программу так, чтобы она выводила на экран пользователя количество предыдущих к ней обращений. Воспользуйтесь для этого скрытыми полями.
8. Напишите программу - калькулятор, которая позволит передать пользователю передать два числа и указать операцию, выполняемую над ними.
9. А на сервере организуйте вычисление результата и выведите его.
10. В отчете отметьте результаты выполненных вами действий и листинги программ.

Анализ полученных результатов

Проанализируйте результаты работы и подумайте какие дополнительные функции можно реализовать в рамках предложенных заданий.

Контрольные вопросы

1. Что такое клиент-серверное взаимодействие?
2. Что такое глобальные переменные?
3. Чем отличается работа метода POST от метода GET?
4. Что такое форма и зачем она нужна?