

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
Учреждение образования
«Гомельский государственный технический университет имени П.О. Сухого»
Кафедра «Информационные технологии»

Лабораторный практикум
старшего преподавателя Стефановского И.Л.
по дисциплине

«Программирование в Internet»

для специальности 1–40 05 01 – "Информационные системы и технологии
(по направлениям)»,
направления специальности 1 40 05 01– 01 – «Информационные системы и
технологии (в проектировании и производстве)»

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Лабораторная работа №1 «Программирование простых скриптов на JavaScript»	4
2. Лабораторная работа №2 «Применение JavaScript для обработки форм и создания динамических страниц»	8
3. Лабораторная работа №3 «Программирование свойств окна браузера, работа с элементами формами»	10
4. Лабораторная работа №4 «Применение JavaScript для динамического управления документом»	14
5. Лабораторная работа №5 «Программирование простых скриптов на PHP»	16
6. Лабораторная работа №6 «Методы передачи данных сценариям. Обработка форм»	27
Список использованных источников	29

ВВЕДЕНИЕ

Дисциплина «Программирование в Internet» знакомит студентов с основными концепциями, технологиями и тенденциями развития глобальной компьютерной сети Интернет, а также с технологией разработки приложений, ориентированной на работу в Интернет. Практическая часть курса посвящена получению навыков решения задач Интернет – программирования.

Цель дисциплины – изучение современных методов программирования приложений, использующих в своей работе среду Internet. А так же создания интернет сайтов, наполненных динамически изменяющимся контентом.

Задачи дисциплины – овладение технологией создания динамических Web-страниц, обучение программированию в Internet на стороне клиента и сервера.

Лабораторный практикум предназначен для выполнения лабораторных работ по курсу «Компьютерная графика» для студентов специальностей 1-40 01 02 «Информационные системы и технологии» направления специальности 1-40 01 02-01 – «Информационные системы и технологии (в проектировании и производстве)».

Цель данного практикума:

1. Научиться создавать динамические Web-страницы с использованием JavaScript;
2. Осуществлять управление внешним видом Web-страницы с использованием JavaScript;
3. Изменять структуру Web-страницы с использованием JavaScript;
4. Осуществлять проверку правильности заполнения форм на Web-странице.
5. Проектировать и создавать динамические веб-сайты с использованием JavaScript и PHP;
6. Формулировать и решать задачи проектирования веб – ориентированных Интернет/Инtranет – приложений с использованием современных технологий клиентского и серверного программирования;
7. Обрабатывать данные форм;
8. Осуществлять авторизация доступа с помощью сессий;
9. Использовать шаблоны в PHP;
10. Использовать элементы предобработки на стороне клиента;

Лабораторная работа №1 «Программирование простых скриптов на JavaScript»

Цель работы: изучить основы программирования на JavaScript, основные операторы и функции языка JavaScript

Краткие теоретические сведения

Запуск JavaScript

Что необходимо сделать, чтобы запускать скрипты, написанные на языке JavaScript? Вам понадобится браузер, способный работать с JavaScript - например Netscape Navigator (начиная с версии 2.0) или Microsoft Internet Explorer (MSIE - начиная с версии 3.0). С тех пор, как оба этих браузера стали широко распространены, множество людей получили возможность работать со скриптами, написанными на языке JavaScript. Несомненно, это важный аргумент в пользу выбора языка JavaScript, как средства улучшения ваших Web-страниц.

При этом, возможно, Вы обнаружите, что много хороших средств диалога можно создать, пользуясь лишь командами HTML. Чтобы получить дополнительную информацию о языке HTML, лучше всего инициировать поиск по ключевому слову 'html' на поисковом сервере Yahoo.

Размещение JavaScript на HTML-странице

Код скрипта JavaScript размещается непосредственно на HTML-странице. Чтобы увидеть, как делается, давайте рассмотрим следующий простой пример:

```
<html>
<body>
<br>
Это обычный HTML документ.
<br>
  <script language="JavaScript">
    document.write("А это JavaScript!")
  </script>
<br>
Вновь документ HTML.
</body>
</html>
```

С первого взгляда пример напоминает обычный файл HTML. Единственное новшество здесь - конструкция:

```
<script language="JavaScript">
  document.write("А это JavaScript!")
```

</script>

Это действительно код JavaScript. Чтобы видеть, как этот скрипт работает, запишите данный пример как обычный файл HTML и загрузите его в браузер, имеющий поддержку языка JavaScript. В результате Вы получите 3 строки текста:

Это обычный HTML документ.
А это JavaScript!
Вновь документ HTML.

Я должен признать, что данный скрипт не столь полезен - то же самое и более просто можно было бы написать на "чистом" языке HTML. Я всего лишь хотел продемонстрировать Вам тэг признака <script>. Все, что стоит между тэгами <script> и </script>, интерпретируется как код на языке JavaScript. Здесь Вы также видите пример использования инструкции document.write() - одной из наиболее важных команд, используемых при программировании на языке JavaScript. Команда document.write() используется, когда необходимо что-либо написать в текущем документе (в данном случае таким является наш HTML-документ). Так наша небольшая программа на JavaScript в HTML-документе пишет фразу "А это JavaScript!".

Список математических функций приведен в таблице 1.1.

Таблица 1.1 - Список математических функций

abs (<i>число</i>) — модуль (абсолютное значение) числа;
acos (<i>число</i>) — арккосинус числа;
asin (<i>число</i>) — арксинус числа;
atan (<i>число</i>) — арктангенс числа;
atan2 (<i>х</i> , <i>у</i>) — угол в полярных координатах точки;
ceil (<i>число</i>) — округление числа вверх до ближайшего целого;
cos (<i>число</i>) — косинус числа;
exp (<i>число</i>) — число <i>e</i> в степени <i>число</i> ;
floor (<i>число</i>) — округление числа вниз до ближайшего целого;
fmod (<i>число1</i> , <i>число2</i>) — возвращает дробный остаток от деления первого числа на второе. Остаток имеет такой же знак, что и <i>число 1</i> .
log (<i>число</i>) — натуральный логарифм числа;
log10 (<i>число</i>) — десятичный логарифм числа;
max (<i>число1</i> , <i>число2</i> , ..., <i>числоN</i>) — большее из чисел в списке;
max (<i>массив_чисел</i>) — большее из чисел в массиве;
min (<i>число1</i> , <i>число2</i> , ..., <i>числоN</i>) — меньшее из чисел в списке;
min (<i>массив_чисел</i>) — меньшее из чисел в массиве;
pow (<i>число1</i> , <i>число2</i>) — <i>число1</i> в степени <i>число2</i>
rand (<i>число1</i> , <i>число2</i>) — случайное число между заданными числами;
round (<i>число</i>) — округление числа до ближайшего целого;
sin (<i>число</i>) — синус числа;

sqrt (число) — квадратный корень из числа;
srand (число) — устанавливает начальное значение генератора случайных чисел в указанное число; если параметр не указан, то начальное число выбирается случайно;
tan (число) — тангенс числа.

Задание на лабораторную работу

В соответствии с вариантом написать JavaScript для вычисления значения функции. Ввод/вывод исходных данных осуществляется через диалоговые окна.

В отчете привести текст HTML документа со скриптом, скриншоты диалоговых окон. Варианты приведены в таблице 1.2.

Таблица 1.2- Варианты заданий

Вариант	Вид функции	Вариант	Вид функции
1	$b = \frac{1 + \cos^2(x+z)}{ x^3 - 2y^2 }$	16	$b = x + \frac{\sqrt[3]{zy}}{y + \cos x}$
2	$b = \frac{\ln^2 z }{\sqrt[3]{ x + y }}$	17	$b = \lg\left(\sqrt{e^{x-y} + x^{ y } + z}\right)$
3	$b = \frac{y^3}{x + y^3 \cos^2 z}$	18	$b = 1 + \frac{x^2 + 1}{3 + y^2} + \sin 2z$
4	$b = \sqrt{x + \sqrt[4]{ y }} + \cos^2 z$	19	$b = \cos x + \cos y + 2 \sin^2 z$
5	$b = \frac{\sqrt[3]{e^{\sin x}} \cdot \cos y}{z^2 + 1}$	20	$b = \frac{\ln(y^3)(z - x/2)}{2 \cos^2 x}$
6	$b = z(\lg y - e^{-(x+3)})$	21	$b = \sqrt{10(\sqrt[3]{z} + x^{(y+2)})}$
7	$b = x - y (\sin^2 z + \lg z)$	22	$b = (\sin z)^2 + x + y $
8	$b = \sqrt{y + \sqrt[3]{x}} - 1 + 2z$	23	$b = e^{2z} - \sqrt[3]{y x }$
9	$b = x(\lg z + \cos^2 y)$	24	$b = e^{(x-1)} + \sin y$
10	$b = e^{ x-y } (\lg^2 z + 1)^x$	25	$b = \sqrt{ z } e^{-(y+x/2)}$
11	$b = \cos^2 z + \lg 2x + y $	26	$b = \frac{4y^2 e^{2x} \sin^2 z}{3z^3 + \ln x}$
12	$b = 5 \lg z - 4y^2 + xy $	27	$b = \frac{\sqrt{y \ln x} - z x^2}{1 + \lg^2 x^2} x$

Вариант	Вид функции	Вариант	Вид функции
13	$b = (z - x) \frac{y - \ln z}{1 + (y - x)^2}$	28	$b = \frac{\lg(y + \sqrt{z + x^2})}{y + x^2}$
14	$b = y^z + \sqrt{ x + y }$	29	$b = \frac{x^2 + 4}{\sin^2 z^2 + x/2} y$
15	$b = \frac{\lg(\sqrt{x} + \sqrt{y} + 2)}{ 2z }$	30	$b = \frac{\sin x + \sqrt{ z - y }}{y(x - 2) + x^2}$

Контрольные вопросы

1. В каком из контейнеров верно используется обработчик события onSelect?
2. В каком из контейнеров используется обработчик события onClick?
3. В каком из контейнеров используется обработчик события onMouseOver?
4. В чем ошибка ниже приведенной синтаксической конструкции?
5. внутри тэгов <script> и </script>?
6. Восстановить значения умолчания в полях формы можно только по кнопке reset?
7. Где отобразится результат работы сценария ?
8. Где расположить сценарий для вывода нового окна в коде странице?
9. Дано а = "м", укажите, когда пользователь увидит сообщение
10. Для чего изображения размещают в таблицах?

Лабораторная работа №2 «Применение JavaScript для обработки форм и создания динамических страниц»

Цель: изучить применение языка JavaScript для обработки форм и создания динамических страниц.

Краткие теоретические сведения

Теоретические сведения смотри в материалах к лекциям.

Задание на лабораторную работу

В соответствии с вариантом создать форму для ввода данных. Страница должна содержать следующие визуальные элементы:

1. заголовок;
2. текст поясняющий назначение формы;
3. форму ввода данных, в соответствии с вариантом. Типы элементов для ввода данных подобрать самостоятельно (не менее 3 различных);
4. кнопки – «**ОТПРАВИТЬ**», «**ОЧИСТИТЬ**»;
5. информацию о создателе страницы – **ФИО, группа, e-mail**.

Для оформления страницы использовать стилевые таблицы (во внешнем файле). Но можно дополнить страницу визуальными эффектами.

The diagram illustrates a web page layout within a rectangular frame. At the top, the text 'Заголовок' (Header) is centered. Below it, 'Пояснительный текст' (Explanatory text) is centered. Underneath the text is a dashed rectangular box containing the text 'Форма ввода данных' (Data input form). At the bottom of the frame, a solid horizontal bar contains the text 'Контактная информация' (Contact information).

Рисунок 2.1. – Примерный вид страницы

К каждому полю формы добавить кнопку «**Пояснение**», при нажатии на которую открывается окно с поясняющей информацией и или картинкой, кнопкой «**ЗАКРЫТЬ**».

При нажатии на кнопку «**ОТПРАВИТЬ**», данные формы проверяются на правильность заполнения – все поля заполнены, числовые поля содержат допустимые значения. При не соответствии данных одного или нескольких полей, на экран должно выводиться окно в котором указываются название и краткий текст поясняющий требования к вводимым в поле данным. Если данные введены верно, то данные отправляются после подтверждения в диалоговом окне.

В отчете представить: скриншоты основного и вспомогательных информационных окон, листинг страницы, данные из строки адреса после нажатия кнопки «**ОТПРАВИТЬ**».

Таблица 2.1. – Варианты форм ввода данных

Вариант	Описание
1	Цех, участок, Ф.И.О. Объем выполненной работы
2	УДК, Ф.И.О. автора, Наименование, Количество
3	Номер поезда, Наименование, Место отправления, Место прибытия, Категория (скорый.)
4	Организация, Ф.И.О., Год рождения, Занятое место
5	Страна, Площадь, Население, Континент, Столица
6	Марка автомобиля, Номер, Цвет, Год выпуска, Пробег
7	Порода собаки, Кличка, Отец, Мать, Дата рождения, ФИО хозяина
8	Вид овощей, Название сорта, Дата посадки, Дата уборки, Урожай
9	Дисциплина, Объем лекций, Объем лабораторных, Вид контроля, Группа
10	Дата, Температура, Давление, Облачность, Направление ветра
11	Наименование, Фирма, Стоимость, Количество, Дата
12	Название группы, Страна, Альбом, Дата выпуска, Число продаж
13	Название вершины, Высота, Страна, Год покорения, Количество восхождений
14	Ф.И.О., Год рождения, Рост, Вес, Группа крови
15	Ф.И.О., Область деятельности, Год рождения, Страна, Число публикаций

Содержание отчета:

- Тема и цель работы.
- Контрольные задания, коды скриптов, результаты работы (более подробные требования приведены в практическом задании).

Контрольные вопросы.

1. Что содержится в свойстве aLinkColor объекта document?
2. Что содержится в свойстве appName объекта navigator?
3. Что содержится в свойстве appVersion объекта navigator?
4. Что содержится в свойстве bgColor объекта document?
5. Что содержится в свойстве fgColor объекта document?
6. Что содержится в свойстве length объекта history?
7. Что содержится в свойстве linkColor объекта document?
8. Что содержится в свойстве title объекта document?
9. Что содержится в свойстве userAgent объекта navigator?
10. Что содержится в свойстве vLinkColor объекта document?

Лабораторная работа №3 «Программирование свойств окна браузера, работа с элементами формами»

Цель: изучить основные свойства и методы объектов Window способы создания и работы с окнами, обращения к элементам форм

Краткие теоретические сведения

Теоретические сведения смотри в материалах к лекциям.

Задание на лабораторную работу

Разработать динамическую страницу для проведения тестирования. Главная страница должна содержать следующие визуальные элементы:

1. заголовок;
2. описание тестового задания (тематика, правила ответа на вопросы);
3. форму для ввода данных о тестируемом (ФИО – 3 поля, факультет – переключатель; группа – список, e-mail);
4. список вопросов;
5. Кнопка – «Результат».

Список вопросов оформляется в виде кнопок, при нажатии на которые динамически формируется окно с вопросом и формой для ответа. Для ввода ответа, должны **обязательно** использоваться следующие элементы – поля ввода, списки и переключатели (в зависимости от вопроса). По нажатию на кнопку «Результат», формируется окно с результатами прохождения теста – личные данные, вопрос с вариантом пользователя и верным, % правильных ответов.

Данные вводимые пользователем должны обязательно проверяться на правильность заполнения. При не допустимых данных, необходимо выводить предупреждение, а поле очищать.

Для хранения информации о вопросе рекомендуется следующая структура:

1. вопрос;
2. варианты ответа;
3. тип элемента для ввода ответа;
4. ответ пользователя;
5. другие поля по необходимости.

Базу вопросов удобно задать в виде массив объектов, каждый экземпляр которого задает структуру описывающую вопрос.

Пример 1. Пример базу для хранения информации о сотрудниках. Основные данные: идентификатор – **id**, фамилия – **fam**, имя – **nam**, должность – **job**:

```
//конструктор объекта задающего структуру  
function man(id,fam,nam,job)  
{
```

```

this.id=id
this.fam=fam
this.nam=nam
this.job=job
}
// массив записей справочника
data = new Array()
data[0]= new man("1","Иванов","Иван","Директор")
data[1]= new man("2","Петров","Петр","Водитель")
data[2]= new man("3","Сидоров","Игнат","Программист")
data[3]= new man("4","Федоров","Федор","Сторож")

```

Пример 2. Пример передачи значения параметра из динамически формируемого окна (**способ 1**). Текущее значение цвета хранится в переменной **«global_color»**. Для изменения ее значения формируется скриптом **«colorSetup()»**, формируется динамическое окно, с формой для ввода значения.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN" >
<HTML>
<HEAD>
<SCRIPT>
global_color="red"; // Глобальная переменная

// скрипт для формирования динамического окна и ввода нового значения
цвета
function colorSetup()
{
    msgWindow=window.open('color', "width=300, height=200");
    msgWindow.document.write('<FORM Name=fr1><INPUT name=colorIn
type=text value="" size=10>')
    msgWindow.document.write("<INPUT TYPE=button VALUE='Установить!'
onclick='opener.global_color=fr1.colorIn.value;'>")
    msgWindow.document.write("<INPUT TYPE=button onclick='window.close();'
VALUE='Закрыть окно'></FORM>")
    msgWindow.document.close()
}
</SCRIPT>

</HEAD>
<BODY>
<FORM Name=fr>
    <BR><INPUT TYPE="button" VALUE="Color!!!" onclick="colorSetup()">
    <INPUT TYPE="button" VALUE="Test Color!!!"
onclick="alert(global_color)">

```

```
</FORM>
</BODY>
</HTML>
```

Пример 3. Пример передачи значения параметра из динамически формируемого окна (способ 2). Текущее значение цвета хранится в переменной «global_color». Для изменения ее значения формируется скриптом «colorSetup()», формируется динамическое окно, с формой для ввода значения:

```
<HTML>
<HEAD>
<SCRIPT language="javascript">
// Глобальные переменные
global_color="red"; // Глобальная переменная
// скрипт для формирования динамического окна и ввода нового значения
// цвета
function colorSetup() //вывод формы изменения
{
    wind=window.open("");
    wind.document.write("<form name=fr1>" +
        "Введите цвет фона <INPUT type=TEXTBOX name=colorIn
size='10'/><br/>" +
        "<INPUT TYPE=button Value='Закрыть' OnClick='opener.zamena();' />" +
        "</form>");
    wind.document.close();
}

function zamena()
{
    if(wind.document.fr1.colorIn.value != "") global_color =
wind.document.fr1.colorIn.value;
    wind.close();
}
</SCRIPT>
</HEAD>

<BODY>
<FORM Name=fr>
    <BR><INPUT TYPE="button" VALUE="Color!!!" onclick="colorSetup()">
    <INPUT TYPE="button" VALUE="Test Color!!!"
onclick="alert(global_color)">
</FORM>
</BODY>
</HTML>
```

Содержимое отчета

В отчете представить: скриншоты основного окна, окон с различными формами для ответов на вопросы, окно с результатами тестирования. Также представить листинги скриптов с комментариями.

Выполнить описание всех глобальных переменных, объектов и функций. Для переменных – назначение, тип. Для объектов – назначение, описание полей, методов. Для функций – назначение, описание списка параметров, результат.

Контрольные вопросы

1. Что называется массивом?
2. Что называется методом?
3. Что называется объектом?
4. Что называется переменной?
5. Что называется свойством?
6. Что называется событием?
7. Что называется циклом?
8. Что положено в основу JavaScript-мультипликации?
9. Что представляет собой тип данных "литерал"?
10. Что представляют собой формы?

Лабораторная работа №4 «Применение JavaScript для динамического управления документом»

Цель: изучить применение языка JavaScript для динамического создания элементов с использованием интерфейса иерархии узлов

Краткие теоретические сведения

Теоретические сведения смотри в материалах к лекциям.

Задание на лабораторную работу

В соответствии с вариантом реализовать программный интерфейс для динамического создания заданного элемента страницы (табл. 4.1). Создаваемый элемент должен добавляется в заданный элемент страницы, определенный блоком **div** или **тегом**, с идентификатором **id**. При необходимости можно использовать стилевые таблицы.

Таблица 4.1. – Динамические элементы

Вариант	Описание
1	Информационная панель – кольцо, с кнопками прокрутки влево, вправо, автоматическая смена по времени.
2	Электронные часы, каждая цифра задается динамически загруженной картинкой. Входными параметрами являются набор изображений и вид часов (12 или 24). Возможны дополнительные параметры.
3	Слайдшоу с оценками фотографий. Следующая картинка подгружается в фоновом режиме. Идентификатор картинки и выставленная оценка передаются на сервер по нажатию на кнопку.
4	Гистограмма задаваемая масштабирование картинки. Исходными данными являются: подпись гистограммы, значение и вид столбца, подписи.
5	Рекламная панель. Исходными данными являются: заголовок, описание, рисунок, ссылка.
6	Списки. Исходными данными являются: идентификатор элемента списка, включая возможность графического идентификатора.
7	Календарь на заданный месяц, заданного года. заданный день выделить
8	Графическое меню с изменением изображения при подводе указателя. Исходными данными являются: наборы динамически подгруженных картинок, ссылки, активный элемент меню, тип – вертикальное или горизонтальное.
9	Вертикальное меню – каталог ссылок. Исходными данными являются: заголовок, ссылки, стилевое оформление, вид обрамления.
10	Гистограмма задаваемая блочным элементом div. Исходными данными являются: подпись гистограммы, значение и цвет столбца, подписи.

11	Голосование. Исходными данными являются: вопросы, элемент для выбора ответа – список, раскрывающийся список, элемент выбора.
12	Прогноз погоды. Исходными данными являются: Город, температура, облачность, ветер.
13	Таблица. Исходными данными являются: подпись таблицы, число строк, столбцов, шапка таблицы, стили для оформления таблицы, шапки, четных (или не четных) строк, данные ячеек.
14	Форма с заданными элементами и подписями.
15	Горизонтальное текстовое меню с картинками. Исходными данными являются: подписи, рисунки, ссылки, стилевое оформление.

Контрольные вопросы

1. Каким должен быть обработчик события кнопки onClick, чтобы корректно вывести содержимое текстового поля fname и длину содержимого текстового поля lname в отдельное окно?
2. Каким должен быть обработчик события кнопки onClick, чтобы корректно вывести содержимое текстового поля muname в отдельное окно?
3. Каким должен быть обработчик события кнопки onClick, чтобы корректно вывести длину содержимого текстового поля muname в отдельное окно?
4. Каким из операторов if или else обрабатывается нажатие кнопки "Отмена" при использовании метода confirm?
5. Каким из операторов if или else обрабатывается нажатие кнопки "ОК" при использовании метода confirm?
6. Каким из перечисленных способов можно отправить данные из формы на сервер?
7. Каким из перечисленных способов можно отправить данные из формы на сервер?
8. Каким образом можно обратиться к 3-й картинке документа ?
9. Каким образом можно обратиться к 8-й картинке документа ?
10. Каким образом можно обратиться к 8-й картинке документа ?

Лабораторная работа №5 «Программирование простых скриптов на PHP»

Цель: изучить синтаксис языка PHP, принципы создания простых страниц на стороне сервера, основные операторы языка по работе массивами

Краткие теоретические сведения

Теоретические сведения смотри в материалах к лекциям.

Задание на лабораторную работу

1. Запустить любой ftp клиент. Например TotalCommader – «menu ftp» – «Новое ftp соединение». Задать адрес ftp сервера: it-web:22. Убрать выделение пункта – «Анонимное соединение»- Рис. 5.1.

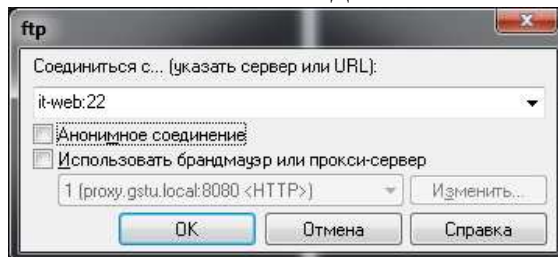


Рис. 5.1-Соединение с ftp-сервером

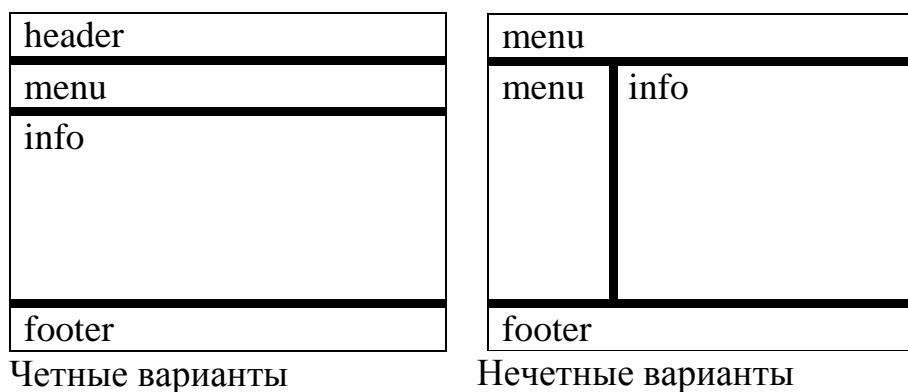
Вести имя пользователя и пароль: **it-domain\логин**. Если подключение прошло удачно – вы находитесь в корневой папке вашего сайта.

2. Создать титульную страницу (**index.html**) на которой разместить: **ФИО студента**, группа, ссылку на первую лабораторную работу. В последующем на эту страницу будут добавляться ссылки на другие лабораторные работы. Перекопируйте созданную страницу в корневую папку сайта. Откройте браузер и в строке адреса задайте путь к сайту: **http://it-web/it-domain/логин**
3. В корневой папке сайта создать папку – «**Laba01**», в которой разместить файл **index.php** следующего содержания:

```
<?php
echo "Начнем.....!!!";
phpinfo();
?>
```

Протестировать работу скрипта.

4. В соответствии с вариантом, создать макет сайта заданной структуры (рисунок 5.2). В вариантах **2, 3, 6, 7, 10, 11, 14, 15** – макет задается с помощью **CSS** позиционирования, в остальных таблицей. Для оформления содержимого страницы используется **CSS**.
- 5.



Четные варианты

Нечетные варианты

Рисунок 5.2 - Структура сайта

Макет страницы выполняется шириной – **800px**, с выравниванием – по центру и содержит следующие основные блоки:

1. header – заголовок сайта, содержит логотип (изображение) и название;
2. menu – меню навигации;
3. info – основной раздел в котором располагается все информативная часть сайта;
4. footer – содержит ФИО студента, группу, дату создания, email и др.

При табличном способе задания макета блоки рекомендуется задавать отдельными таблицами которые включаются в основной макет.

4. Разбить созданный макет сайта на блоки header, menu, footer и сохранить их в файлах с расширениями php в папке – Laba01/templates. Создать в папке – Laba01, файл index.php в который, с помощью операторов включения require, include добавить ранее созданные блоки. Проверить работу.

5. Создать в папке – Laba01/templates, страницы наполнения сайта следующего содержания:

1. main.php – содержит цель, задание к лабораторной работе, информацию о студенте – ФИО, группа, фотография и т.д;
2. page1.php – краткое описание содержимого страницы, исходные данные в виде таблицы в соответствии с вариантом задания. Заголовок и четные строки вывести различным оформлением с применением стилевых таблиц. Исходные данные задаются в виде массива в отдельном файле – Laba01/templates/array.php;
3. page2.php – краткое описание содержимого страницы, результаты сортировки данных по заданным критериям – полю и ключу (выбрать самостоятельно). Выполнить сортировку по двум критериям с использованием функции сортировки заданной пользователем. Поля для сортировки задаются непосредственно в самом скрипте;
4. page3.php – данные отфильтрованные по заданным нескольким критериям (выбрать самостоятельно). Условия фильтров задаются непосредственно в скрипте.

Все страницы оформляются только с использованием CSS.

Таблица 5.3. – Варианты базы данных

Вариант	Описание
1	Цех, участок, Ф.И.О. Объем выполненной работы
2	УДК, Ф.И.О. автора, Наименование, Количество
3	Номер поезда, Наименование, Место отправления, Место прибытия, Категория (скорый.)
4	Организация, Ф.И.О., Год рождения, Занятое место
5	Страна, Площадь, Население, Континент, Столица
6	Марка автомобиля, Номер, Цвет, Год выпуска, Пробег
7	Порода собаки, Кличка, Отец, Мать, Дата рождения, ФИО хозяина
8	Вид овощей, Название сорта, Дата посадки, Дата уборки, Урожай
9	Дисциплина, Объем лекций, Объем лабораторных, Вид контроля, Группа
10	Дата, Температура, Давление, Облачность, Направление ветра
11	Наименование, Фирма, Стоимость, Количество, Дата
12	Название группы, Страна, Альбом, Дата выпуска, Число продаж
13	Название вершины, Высота, Страна, Год покорения, Количество восхождений
14	Ф.И.О., Год рождения, Рост, Вес, Группа крови
15	Ф.И.О., Область деятельности, Год рождения, Страна, Число публикаций

Содержимое отчета

Теоретическая часть отчета должна содержать описание все использованных операторов языка РНР. В отчете также приводится код всех страниц с описанием назначения каждой. Также привести содержимое CSS файла и вид нескольких страниц сайта.

Примеры

Составим макет страницы сайта:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Пример 1</title>
<link href="default.css" rel="stylesheet" type="text/css" >
</head>
<body>
<table id="maket" cellspacing="2" cellpadding="2" border="1">
<tr>
<td colspan="2">
<h1>Пример сайта</h1></td>
<tr>
<td width="20%">
```

```

    <b>=Меню=-</b><br>
    <a href="index.php">Главная</a><br>
    <a href="page1.php">Страница 1</a><br>
    <a href="page2.php">Страница 2</a><br></td>
</td>
Контент
</td>
</tr>
<tr>
<td colspan="2" align=center>
    <B>copywrite Иванов И.И., 2010</b> </td>
</tr>
<table>
</body>
</html>

```

default.css

```

* {
    margin: 0;
    padding: 0;
}
body {
    font: normal small Arial, Helvetica, sans-serif;
    color: #999999;
    font-size: 1.0em;
}
h1, h2{
    margin: 0;
    color: #11A6D4;
}
h2 {
    margin-bottom: .5em;
    font-size: 1.8em;
}
a {
    color: #EA672E;
}

a:hover {
    text-decoration: none;
    color: #11A6D4;
}
#maket {
    width: 800px;
    margin: 0 auto;
}

```

```
}
```

Пример создания с использование блочной модели построения.

Разобьем макет страницы на блоки – header, menu, footer. Код, соответствующий каждому блоку, описывающему элемент страницы, поместим в отдельный файл с расширением – php. Сборку страницы выполним в отдельном файле – main_blok.php.

header.php

```
<h1>Пример сайта</h1>
```

menu.php

```
<b>--Меню--</b><br>
<a href="index.php">Главная</a><br>
<a href="page1.php">Страница 1</a><br>
<a href="page2.php">Страница 2</a><br>
```

footer.php

```
<B>copywrite Иванов И.И., 2010</b>
```

main_blok.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Пример 1</title>
<link href="default.css" rel="stylesheet" type="text/css" >
</head>
<body>
<table id="maket" cellspacing="2" cellpadding="2" border="1" >
<tr>
<td colspan="2">
<?php include ("header.php");?>
</td>
<tr>
<td width="20%">
<?php include ("menu.php");?>
</td>
<td>
<?php echo $content;?>
</td>
</tr>
<tr>
<td colspan="2" align=center>
<?php include ("footer.php");?>
</td>
```

```

</tr>
<table>
</body>
</html>

```

Содержательная часть каждой страницы сайта должна помещаться в переменную \$content. Таким образом, для создания страницы сайта необходимо разместить в нутрии каждой код, создающий представление страницы в переменной \$content и включить в него файл – main_blok.php описывающий макет страницы. Поместим все вспомогательные файлы – header, menu, footer, main_blok.php в папку **templates**. Страницы сайта будут иметь следующий вид:

index.php

```

<?php
$content=""
<h2>Главная страница</h2>
<p>Добро пожаловать на наш сайт!...</p>
";

```

```

#получаем ip-адрес посетителя который пытается посмотреть страницу.

```

```

$ip=getenv('REMOTE_ADDR');
$content.= '<b>IP<b>: '.$ip.'<br>';

```

```

#узнаем каким браузером пытаются посмотреть страницу

```

```

$browser=getenv('HTTP_USER_AGENT');
$content.= '<b>BROWSER</b>: '.$browser.'<br>';

```

```

#узнаем как называется наш скрипт

```

```

$name=$_SERVER['PHP_SELF'];
$content.= '<b>Название скрипта<b>: '.$name.'<br>';

```

```

#Узнаем на каком сервере выполняется скрипт

```

```

$host=$_SERVER['SERVER_NAME'];
$content.= '<b>SERVER<b>: '.$host.'<br>';

```

```

include ("templates/main_blok.php");
?>

```

Результат работы показан на рис. 5.3.

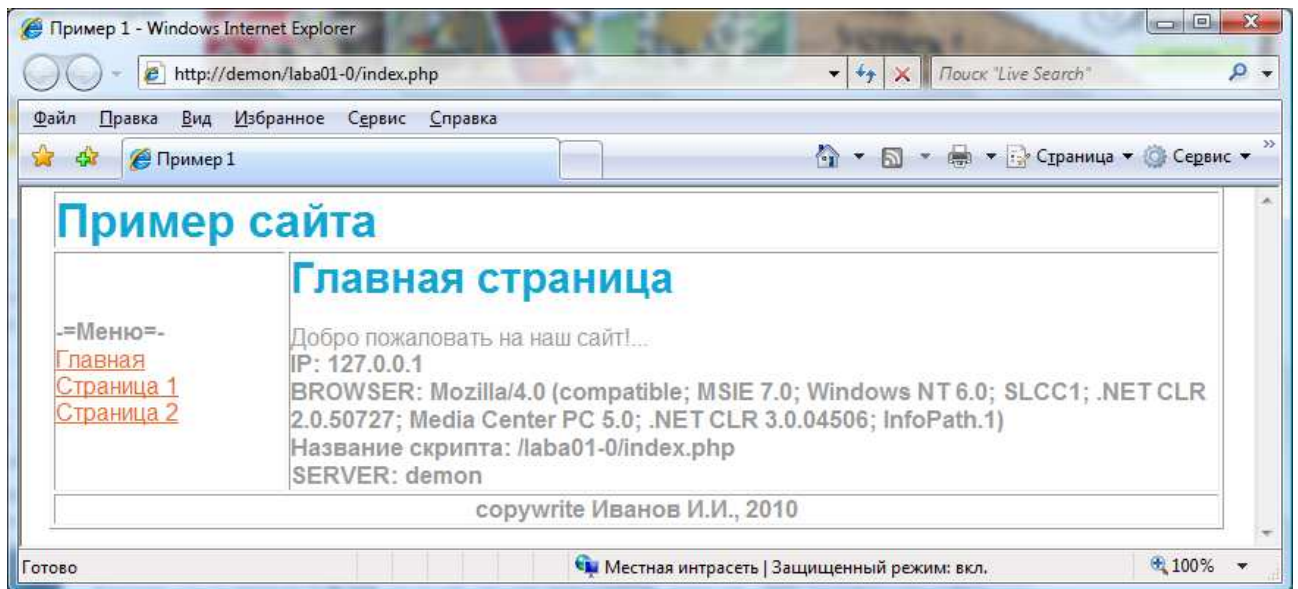


Рис. 5.3 – Результат работы скрипта index.php

page1.php

```
<?php
$content="
<h2>Страница #1</h2>
<p>Добро пожаловать на наш сайт!...</p>
";
include ("templates/main_blok.php");
?>
```

Результат работы показан на рис. 5.4.

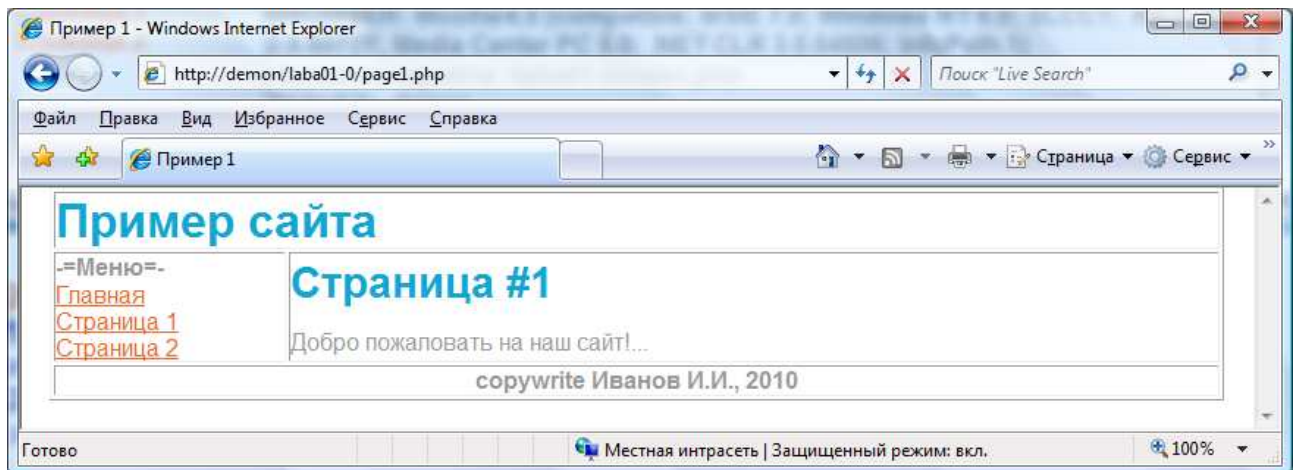


Рис. 5.4 – Результат работы скрипта page1.php

Page2.php

```
<?php
$content="
<h2>Страница #2</h2>
<p>Добро пожаловать на наш сайт!...</p>
";
```

```
include ("templates/main_blok.php");  
?>
```

Пример создания с использование блочной модели построения.

Разобьем макет страницы на блоки – header, menu, footer следующего содержания:

header.php

```
<table id="maket" cellspacing="2" cellpadding="2" border="1">  
<tr>  
<td colspan="2">  
  <h1>Пример сайта</h1>  
</td>
```

menu.php

```
<tr>  
<td width="20%">  
  <b>==Меню==</b><br>  
  <a href="index.php">Главная</a><br>  
  <a href="page1.php">Страница 1</a><br>  
  <a href="page2.php">Страница 2</a><br>  
</td>  
<td>
```

footer.php

```
</td>  
</tr>  
<tr>  
<td colspan="2" align=center>  
  <b>соруwrite Иванов И.И., 2010</b>  
</td>  
</tr>  
<table>
```

Создадим блоки содержательной части страницы.

page0.php

```
<h2>Главная страница</h2>  
<p>Добро пожаловать на наш сайт!...</p>  
<?php
```

#получаем ip-адрес посетителя который пытается посмотреть страницу.

```
$ip=getenv('REMOTE_ADDR');  
echo '<b>IP<b>: '.$ip.'<br>';
```

#узнаем каким браузером пытаются посмотреть страницу

```
$browser=getenv('HTTP_USER_AGENT');
```

```
echo '<b>BROWSER</b>: '.$browser.'<br>';
```

```
#узнаем как называется наш скрипт  
$name=$_SERVER['PHP_SELF'];  
echo '<b>Название скрипта<b>: '.$name.'<br>';
```

```
#Узнаем на каком сервере выполняется скрипт  
$host=$_SERVER['SERVER_NAME'];  
echo '<b>SERVER<b>: '.$host.'<br>';  
?>
```

page1.php

```
<h2>Страница #1</h2>
```

```
<p>Добро пожаловать на наш сайт!...</p>
```

page2.php

```
<h2>Страница #2</h2>
```

```
<p>Добро пожаловать на наш сайт!...</p>
```

Поместим все созданные вспомогательные файлы в папку **templates**. С помощью операторов включения выполним сборку блоков. Окончательно страницы сайта буду иметь следующий вид:

index.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
```

```
"http://www.w3.org/TR/html4/strict.dtd">
```

```
<html>
```

```
<head>
```

```
<title>Пример 1</title>
```

```
<link href="default.css" rel="stylesheet" type="text/css" >
```

```
</head>
```

```
<body>
```

```
<?php include ("templates/header.php");?>
```

```
<?php include ("templates/menu.php");?>
```

```
<?php include ("templates/page0.php");?>
```

```
<?php include ("templates/footer.php");?>
```

```
</body>
```

```
</html>
```

Результат работы показан на рис. 5.5.

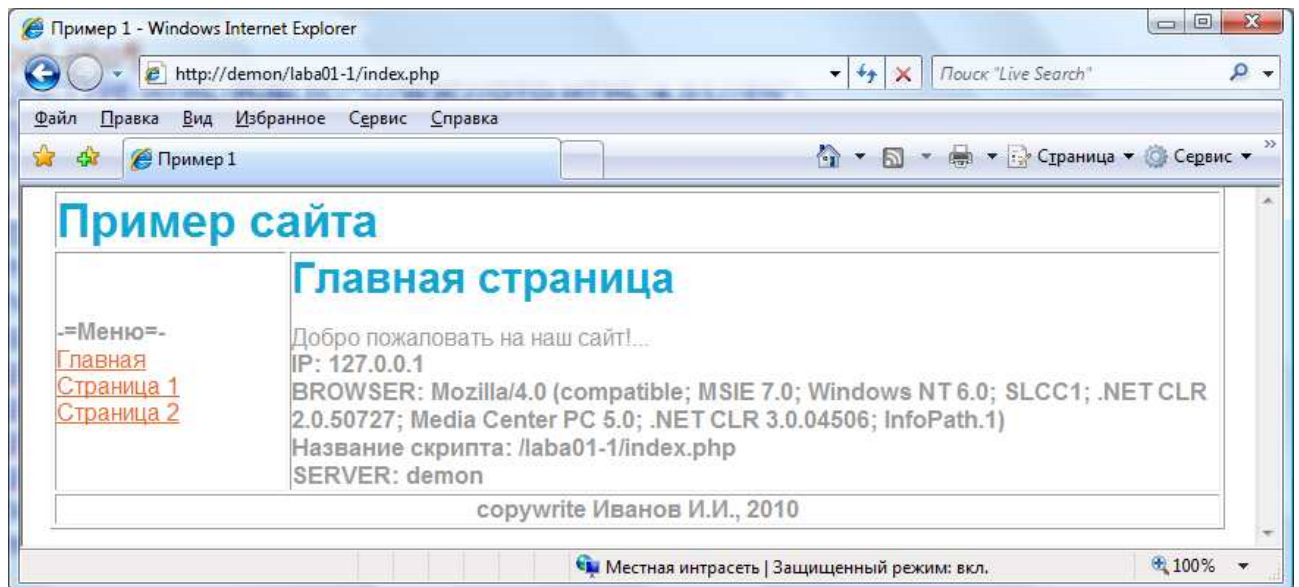


Рис. 5.4 – Пример создания с использованием блочной модели построения

page1.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Пример 1</title>
<link href="default.css" rel="stylesheet" type="text/css" >
</head>
<body>
<?php include ("templates/header.php");?>
<?php include ("templates/menu.php");?>
<?php include ("templates/page1.php");?>
<?php include ("templates/footer.php");?>
</body>
</html>
```

page2.php

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">
<html>
<head>
<title>Пример 1</title>
<link href="default.css" rel="stylesheet" type="text/css" >
</head>
<body>
<?php include ("templates/header.php");?>
<?php include ("templates/menu.php");?>
<?php include ("templates/page2.php");?>
<?php include ("templates/footer.php");?>
```

</body>

</html>

Результат работы показан на рис. 5.6.

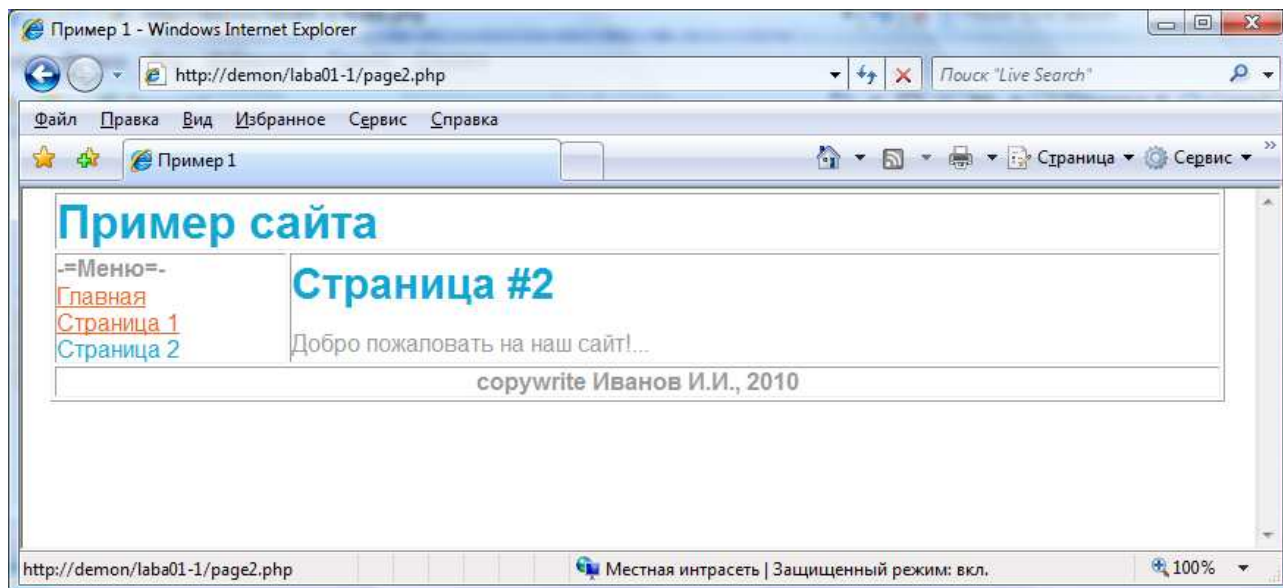


Рис. 5.4 – Пример создания с использование шаблонов

Контрольные вопросы

1. Как встраивается PHP в HTML-код?
2. Как задаются комментарии в языке PHP?
3. Как можно задать массив в языке PHP?
4. Как можно задать строку в языке PHP?
5. Как передаются данные методом GET?
6. Как получить значение константы с именем CONS в языке PHP?
7. Как присваивается значение константе в языке PHP?
8. Как присваивается значение переменной в языке PHP?
9. Как разделяются инструкции в языке PHP?
10. Какие из операторов if записаны правильно с точки зрения синтаксиса?
11. Какие из операторов switch записаны правильно с точки зрения синтаксиса?
12. Какие из утверждений относительно оператора include верны?
13. Какие из утверждений относительно оператора require верны?
14. Какие из циклов while записаны правильно с точки зрения синтаксиса?
15. Какие парадигмы программирования поддерживает PHP?

Лабораторная работа №6 «Методы передачи данных сценариям. Обработка форм»

Цель: изучить методы передачи данных сценариям на стороне сервера, обработку данных запросов на PHP, способы обработки элементов форм.

Краткие теоретические сведения

Теоретические сведения смотри в материалах к лекциям.

Задание на лабораторную работу

1. Создать титульную страницу (**index.html**) добавить ссылку на шестую лабораторную работу.
2. В корневой папке сайта создать папку – «**Laba02**» в которую скопировать сайт созданный в предыдущей лабораторной работе (**Лабораторная по PHP №6.**)
3. Внести следующие **изменения**:
 1. **main.php** – содержит цель, задание к **лабораторной работе №2**, информацию о студенте – ФИО, группа, фотография и т.д;
 2. **page1.php** – Сценарий реализующий формирование формы (в соответствии с вариантом) и ее обработку, без повторного корректно заполненных полей. Введенные данные проверяются на правильность заполнения. В случае ошибки форма выводится повторно, неверно введенные поля очищаются и рядом с ними выводится подсказка. Правильно введенные поля должны сохранять свои значения. Если все поля заполнены верно, пользователю выводится страница с введенными данными. **Форма должна содержать не менее трех различных элементов ввода**;
 3. **page2.php** – краткое описание содержимого страницы, с указанием по каким полям выполняется сортировка, результаты сортировки данных по заданным критериям – полю и ключу. Поле и ключ для сортировки задаются с помощью формы. Выполнить проверку корректности введенных данных. Если данные введены верно, пользователю выводится предупреждение, и таблица не выводится. Поле для сортировки задается **тестовым полем**, а ключ – переключателем **radiobutton**. Исходные данные задаются в виде массива в отдельном файле – **Laba02/templates/array.php**;
 4. **page3.php** – краткое описание содержимого страницы, с указанием по каким полям выполняется выборка данных, данные отфильтрованные по заданным нескольким критериям. Поля для фильтрации задаются формой со **списками**, с сохранением выбранного значения между вызовами сценария. В списки добавить значение – «**ВСЕ**», для отмены фильтра по данному полю.

Все страницы оформляются только с использованием CSS.

Таблица 6.1 – Варианты базы данных

Вариант	Описание
1	Цех, участок, Ф.И.О. Объем выполненной работы
2	УДК, Ф.И.О. автора, Наименование, Количество
3	Номер поезда, Наименование, Место отправления, Место прибытия, Категория (скорый.)
4	Организация, Ф.И.О., Год рождения, Занятое место
5	Страна, Площадь, Население, Континент, Столица
6	Марка автомобиля, Номер, Цвет, Год выпуска, Пробег
7	Порода собаки, Кличка, Отец, Мать, Дата рождения, ФИО хозяина
8	Вид овощей, Название сорта, Дата посадки, Дата уборки, Урожай
9	Дисциплина, Объем лекций, Объем лабораторных, Вид контроля, Группа
10	Дата, Температура, Давление, Облачность, Направление ветра
11	Наименование, Фирма, Стоимость, Количество, Дата
12	Название группы, Страна, Альбом, Дата выпуска, Число продаж
13	Название вершины, Высота, Страна, Год покорения, Количество восхождений
14	Ф.И.О., Год рождения, Рост, Вес, Группа крови
15	Ф.И.О., Область деятельности, Год рождения, Страна, Число публикаций

Содержимое отчета

Теоретическая часть отчета должна содержать описание **использованных операторов языка PHP по обработке формы**. В отчете также приводится код страниц **page1.php, page2.php, page3.php** с описанием назначения каждой. Также привести вид страниц этих страниц.

Контрольные вопросы

1. Чем отличается оператор break от оператора continue?
2. Чем отличается скрипт на языке PHP от скрипта на языке JavaScript?
3. Чем отличаются операторы require и include?
4. Чем цикл while отличается от цикла do..while?
5. Чему будет равна переменная \$result в результате выполнения следующей программы
6. Что мы получим в результате обработки интерпретатором файла index.php?
7. Что нужно сделать для того, чтобы настроить PHP для работы с сервером Apache (PHP устанавливается как серверный модуль)?
8. Что нужно сделать, чтобы получить в браузере строку: "PHP работает: привет"
9. Что такое корень сервера Apache и где он находится по умолчанию (для windows систем)?
10. Что такое пользовательская директория сервера и где она находится по умолчанию (для windows систем)?

Список использованных источников

1. Наварро Э. XHTML. Учебный курс. СПб.: Питер, 2001.
2. Тиге Дж.К. DHTML и CSS для Internet. М.: НТ Пресс, 2005.
3. Зельдман Д. Web-дизайн по стандартам. М.: НТ Пресс, 2005.
4. Дилип Найк Dynamic HTML: Стандарты и протоколы Интернета «Русская редакция» ТОО «Channel Trading Ltd.» 2005 г.
5. Джамса К. Эффективный самоучитель по креативному Web-дизайну. HTML, XHTML, CSS, JavaScript, PHP, ASP, ActiveX. Текст, графика, звук и анимация. М: "ООО ДиаСофтЮП", 2005.
6. В.Олифер, Н.Олифер. Компьютерные сети. Принципы, технологии, протоколы - СПб: "Питер", 2003.
7. Котеров Д. Самоучитель PHP 4. СПб.: БХВ, 2001.
8. Колесниченко Д.Н. Самоучитель PHP 5. СПб.: Наука и техника, 2004.
9. Мельников Д.А. Информационные процессы в компьют.сетях. Протоколы, стандарты, интерфейсы, модели: М: КУДИЦ-ОБРАЗ, 2001, 256 с.
10. Романец Ю.В., Тимофеев П.А., Шаньгин В.Ф. Защита информации в компьют.системах и сетях. Изд. 2-е, перераб., доп. М: Радио и связь, 2001, 376 с.