|  |  |  |
| --- | --- | --- |
| **Технология web-программирования, группа АФТ 05-17, ОЗ АФТ 05-17** | | |
| Дата занятия | Вид занятия | Ссылка на источник |
| 06.04.2020 | Лк | 1. Видеолекция «PHP видео-урок №1. Программирование для начинающих. Введение»   https://www.youtube.com/watch?time\_continue=1&v=pG-XewXQDSk&feature=emb\_logo   1. Торопова О.А. Основы web-программирования. Технологии HTML, DHTML [Электронный ресурс]: учебное пособие / О.А. Торопова, И.Ф. Сытник. – Электрон. текстовые данные. – Саратов: Саратовский государственный технический университет имени Ю.А. Гагарина, ЭБС АСВ, 2012. – 106 c. – 978-5-7433-2606-8. – Режим доступа: http://www.iprbookshop.ru/76493.html 2. Одиночкина С.В. Web-программирование PHP [Электронный ресурс] / С.В. Одиночкина. – Электрон. текстовые данные. – СПб. : Университет ИТМО, 2012. – 79 c. – 2227-8397. – Режим доступа: http://www.iprbookshop.ru/65750.html 3. Электронный ресурс http://weblecture.ru/lecture?num=181 |
| 06.04.2020 | Лб | Выполнить лабораторную работу № 3 «Создание динамических Web-страниц с помощью PHP. Линейные алгоритмы.   1. » Методические указания по выполнению лабораторной работы приведены ниже |

**Лабораторная работа 1.**

**Создание динамических Web-страниц с помощью PHP. Линейные алгоритмы.**

**Цель работы:** формирование умений программирования линейных алгоритмов на PHP.

**Поясняющая информация**

**Веб-сервер** – программа, которая запущена на локальной или удаленной машине, и главное её предназначение – позволять пользователю просматривать веб-документы. Когда в адресной строке вводится адрес сайта, при нажатии клавиши *Enter* браузер формирует некий запрос, направленный удаленному компьютеру — веб-серверу.

**Apache HTTP-сервер** (сокращение от англ. **a patchy server**) – одна из наиболее популярных и распространенных программ, созданных для веб-разработчиков и администраторов интернет-ресурсов. По данным независимых исследователей, Apache установлен на 50% компьютеров всех пользователей HTTP-серверов.

Главными преимуществами Apache являются стабильность, быстродействие и гибкость, и обусловлены они модульной организацией, а также тем, что разработку ведёт открытая группа программистов, хоть и под официальным названием *Apache Software Foundation*.

Apache поддерживает огромное количество операционных систем и аппаратных платформ, и довольно часто его используют под управлением операционной системы Windows. Немаловажно и то, что данный web-сервер распространяется совершенно бесплатно.

В первую очередь необходимо [скачать](http://httpd.apache.org/download.cgi) последнюю версию дистрибутива Apache без поддержки **SSL** и запустить установку. В приветственном окне нужно кликнуть на «*Next*», прочесть лицензионное соглашение разработчика и подтвердить свое согласие с ним.

Затем, в окне установки необходимо заполнить поля так, как показано на картинке ниже, а в поле «*Administrator’s Email Address*» указать адрес своей электронной почты, кликнуть «*Next*» и установить переключатель в положение «*Custom*».

Далее, переходим к следующему шагу процесса «*установка Apache*». По умолчанию веб-сервер производит инсталляцию в папку **C:Program FilesApache Software FoundationApache 2.2**.

Далее необходимо создать директорию **www** на диске **C** и указать её в качестве установочной папки для Apache, кликнуть «*Next*» в этом и «*Install*» в следующем окне. Когда установка закончится, нажмите на «*Finish*».

Если все прошло успешно, в трее должен появиться значок управления веб-сервером Apache. Остановить и перезагрузить его можно, кликнув на значке левой кнопкой мыши.

Кликнув по тому же значку правой кнопкой мыши, можно перейти к различным системным службам операционной системы или открыть монитор Apache.

Для того чтобы проверить работает ли установленный сервер Apache, наберите в адресной строке вашего браузера **http://localhost**

Если появилась страница с воодушевляющей надписью «*It works!*», значит, установка Apache прошла успешно, и он функционирует правильно.

**Базовая настройка веб-сервера**

Хоть Apache и является весьма удобным и исключительно надёжным, не все готовы его использовать как локальный сервер по ряду причин, и основная — отсутствие даже в среде *Microsoft Windows* какого-либо графического конфигуратора, что довольно непривычно для большинства пользователей.

Настройка сервера производится с помощью ручного редактирования конфигурационного файла **httpd.conf**. Однако, вопреки представлениям о непонятности и сложности данного процесса, ничего сложного в данной операции нет по двум причинам: **во-первых**, для того, чтобы сделать из только что установленного Apache сервер, который настроен для приемлемой и комфортной работы, в файле конфигурации нужно поменять совсем немного данных, а **во-вторых** — комментарии **httpd.conf** содержат немало полезной информации, необходимой для того, чтобы разобраться с настройкой.

**Что необходимо?**

* Запустить Apache под платформой Windows 7;
* Хранить в удобной для пользователя директории файлы будущего сайта (например, C:www);
* Не испытывать проблем с кодировкой, а в частности — с отображением кириллицы;
* Возможность работать с несколькими сайтами одновременно.

Чтобы успешно решить эти проблемы, зайдите в папку с установленным Apache, найдите и откройте в подпапке **conf** файл **httpd.conf**. Обратите внимание, что строки, начинающиеся с «*решётки*», являются текстовыми комментариями, а в качестве настроек веб-сервера использованы строки, в начале которых значок «*решётки*» отсутствует.

Для начала нужно выполнить настройку той директории, которая будет использоваться сервером в качестве папки веб-сайта. Найдите в тексте строку, которая начинается с *DocumentRoot*. Данная директива определяет, какая папка будет являться корневой.

Исправьте её на *DocumentRoot* “**C:/www**”. Следует уточнить, что слэши в этом файле должны быть наклонены вправо, а не влево, как, вероятно, уже привыкли пользователи Windows. После вышеуказанных манипуляций, проблема с месторасположением вашего сайта решена.

Далее необходимо установить для директории правила безопасности. В Apache для этой цели используется директива **Directory**:

<Directory>

Options (определяет, какие серверные функции будут доступны)

AllowOverride (определяет, какие директивы из .htaccess могут преобладать над такими же в httpd.conf)

Order (устанавливает определенные правила доступа к серверу)

</Directory>

### Список используемых параметров:

**Options. Возможны следующие варианты:**

* **Includes** – использование SSI разрешено;
* **IncludesNOEXEC** – использование SSI разрешено с ограничениями (не допускаются *#include* и *#exec*);
* **Indexes** – разрешено использование индексных файлов, и если URL-адрес указывает на директорию сайта (например, *www.domain.ru/dir/*), в которой не существует индексного файла, будет показано содержимое данной директории, а если этой опции нет – выдается оповещение о том, что доступ запрещён;
* **ExecCGI** – выполнение скриптов CGI разрешено;
* **FollowSymLinks** – сервер следует по имеющимся символическим ссылкам директории (используется в Unix-системах);
* SymLinksIfOwnerMatch – сервер следует по имеющимся символическим ссылкам директории лишь в том случае, если целевой файл имеет одного владельца со ссылкой;
* **All** – все вышеописанное вместе разрешено;
* **None** – все вышеописанное вместе запрещено;
* **MultiViews** – возможность подбора определенного контента и его выдачи в зависимости от предпочтений браузера (даже если включено все (Options All). Указывается отдельно).

**AllowOverride. Варианты:**

* **AuthConfig** – позволяет использовать директивы для авторизации;
* **FileInfo** – разрешает использование директив для работы с различными типами документов;
* **Indexes** – разрешает использование директив для работы с файлами индексации;
* **Limit** – разрешает использование директив для определения доступа к хосту;
* **Options** – разрешает использование директив для работы с определенными специфическими функциями директорий;
* **All** – все вышеуказанное вместе;
* **None** – ничего из вышеуказанного вместе.

**Order. Варианты:**

* **Deny, Allow** – Deny определяется перед директивой Allow, доступ разрешен по умолчанию, кроме хостов, которые указаны в следующей после Deny from строке;
* **Allow,Deny** – Allow определяется перед директивой Deny, доступ запрещен по умолчанию, кроме хостов, которые указаны в следующей после Allow from строке;
* **Mutual-failure** – разрешен доступ только таких хостов, которые отсутствуют в Deny и присутствуют в Allow.

На основе всего вышесказанного, попробуйте настроить ваш сервер. В файле **httpd.conf** директива Directory по умолчанию существует сразу в двух экземплярах — <Directory/> и <Directory «корневая директория вашего сайта»>. Первый вариант трогать не следует, поэтому во втором установите параметры следующим образом:

<Directory C:/www>

Options Indexes FollowSymLinks

AllowOverride None

Order allow, deny

Allow from all

</Directory>

Говоря другими словами, для папки C:/www и всех её подпапок выбраны следующие параметры:

* Из возможной функциональности имеющегося сервера разрешены индексы в директориях и переходы по символическим ссылкам;
* Возможность параметрического переопределения с помощью файлов .htaccess полностью отсутствует, однако учитывая то, что вы имеете полный доступ к серверу, она не является актуальной – все можно настроить через httpd.conf;
* Доступ к веб-серверу разрешен со всех хостов.

Теперь, сохраните файл **httpd.conf** и перезапустите Apache с помощью Apache Monitor либо командой **apache –k restart** в командной строке. Настройка корневой папки сайта закончена.

Следует проверить, правильно ли вы всё сделали. Создайте простейшую веб-страницу в папке C:www, откройте ваш браузер и введите [http://127.0.0.1/ваша\_созданная\_страница](http://127.0.0.1/%D0%B2%D0%B0%D1%88%D0%B0_%D1%81%D0%BE%D0%B7%D0%B4%D0%B0%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D1%82%D1%80%D0%B0%D0%BD%D0%B8%D1%86%D0%B0). Страницы должна открыться. В ином случае необходимо тщательно проверить все изменения в файле **httpd.conf** на правильность.

Если страница успешно открылась, есть вероятность, что вместо кириллических символов вы увидите нечитаемые символы. Отображаться они могут по двум причинам. **Во-первых**, веб-сервер предоставляет вашему браузеру, который запросил страницу, кодировку по умолчанию. **Во-вторых**, как ни странно, кодировка эта кириллической не является.

Если пресловутый **Internet Explorer** в подобных случаях определяет кодировку из самой страницы, то, к примеру, **Mozilla Firefox** и **Opera** к подобным действиям склонности совершенно не имеют, и способ выставления кодировки в браузере вручную едва ли можно назвать удобным. Следовательно, необходимо настроить Apache на выдачу по умолчанию необходимой кодировки.

Найдите в файле **httpd.conf** строку, которая начинается на **AddDefaultCharset**. Скорее всего, там указана кодировка ISO-8859-1, в которой отсутствуют кириллические символы. Поменяйте **ISO-8859-1** на **windows-1251**, сохраните файл и перезапустите Apache. Теперь корректное отображение русского языка на вашем сайте будет работать в любом браузере.

В сервере Apache достаточно просто настроить работу с несколькими сайтами. Адреса, которые можно использовать для них — **127.0.0.2, 127.0.0.3** и т.д. При этом по сети виден будет только тот, который является сайтом по умолчанию (127.0.0.1), однако для локальной работы это не является критичным. Раздел файла **httpd.conf**, в котором производится настройка всего необходимого для этого, находится в самом конце и именуется **VirtualHosts**.

Для того чтобы добавить дополнительный сайт, создайте директорию, использующуюся в качестве его корневой, к примеру, **C:www2**. Предположив, что сайт будет откликаться на адрес **127.0.0.2**, дайте ему имя **site911** и в самом конце раздела VirtualHosts добавьте следующие строки:

<VirtualHost 127.0.0.2>

ServerAdmin webmaster@site911

ServerName site911

DocumentRoot "C:/www2"

ScriptAlias /cgi/ "C:/www2/cgi/"

ErrorLog "C:/www2/error.log"

CustomLog "C:/www2/custom.log" common

</VirtualHost>

Перезапустите сервер Apache. Набрав в адресной строке браузера **127.0.0.1**, вы попадёте на ваш первый локальный сайт, а набрав **127.0.0.2** – на второй локальный сайт. Следует иметь в виду, что внутри каждого контейнера VirtualHosts могут использоваться абсолютно любые директивы веб-сервера Apache, благодаря чему можно каждый сайт настраивать наидетальнейшим образом.

**PHP-скриптовый язык**

РНР - это действующий на стороне сервера встраиваемый в HTML язык, имеющий синтаксис, близкий к языку Си. Язык РНР дает возможность вставлять в файлы HTML инструкции языка PHP для создания динамического содержания. Эти инструкции обрабатывает препроцессор-интерпретатор РНР и заменяет их тем содержимым, которое производит этот код. Стандартное расширение файла с PHP-программой - php.

**Синтаксис PHP**

PHP-код включаются в html-код в следующем виде:

<?PHP  текст\_кода  ?>

 или

<?PHP  
  текст\_кода;

  ?>

Например, вставка содержимого файла будет выглядеть так:

<?PHP include "ssi/top.php"; ?>

**Коментарии**

PHP поддерживает комментарии 'C', 'C++' и оболочки Unix. Например:

<?php  
echo "This is a test"; // Это однострочный комментарий в стиле c++

/\* Это многострочный комментарий, это ещё одна его строка \*/

echo "This is yet another test";

echo "One Final Test"; # Это комментарий в shell-стиле

?>

echo

<?php echo "Эта информация будет выведена в HTML"; ?>

**Присвоение значений переменным**

Переменные в программах на PHP, отделяются символами $.

$city = "Kazan";

city - переменная

Kazan - значение

**Некоторые операции**

инкремента/декремента;  
++$a Pre-increment Увеличивает $a на 1, затем возвращает $a.

$a++ Post-increment Возвращает $a, затем увеличивает $a на 1.

--$a Pre-decrement Уменьшает $a на 1, затем возвращает $a.

$a-- Post-decrement Возвращает $a, затем уменьшает $a на 1.

арифметические:  
$a + $b Сложение Сумма $a и $b.

$a - $b Вычитание Разность $a и $b.

$a \* $b Умножение Произведение $a и $b.

$a / $b Деление Частное от деления $a на $b.

$a % $b Modulus Целочисленный остаток от деления $a на $b.

Выражения сравнения вычисляются в 0 или 1, означая FALSE или TRUE (соответственно). PHP поддерживает > (больше), >= (больше или равно), == (равно), != (не равно), < (меньше) и <= (меньше или равно). Эти выражения чаще всего используются внутри условных операторов, таких как if.

сравнения:  
$a == $b равно TRUE, если $a равно $b.

$a != $b не равно TRUE, если $a не равно $b.

$a <> $b не равно TRUE, если $a не равно $b.

$a < $b меньше TRUE, если $a строго меньше $b.

$a > $b больше TRUE, если $a строго больше $b.

$a <= $b меньше или равно TRUE, если $a меньше или равно $b.

$a >= $b больше или равно TRUE, если $a больше или равно $b.

строковые:

Имеются две строковые операции. Первая - операция ('.'), которая возвращает объединение из правого и левого аргументов. Вторая - операция присвоения ('.='), которая присоединяет правый аргумент в левому аргументу.

$a = "Hello ";

$b = $a . "World!"; // теперь $b содержит "Hello World!"

$a = "Hello ";

$a .= "World!"; // теперь $a содержит "Hello World!"

***К сдаче лабораторной предоставляются:*** *работающие страницы на сервере.*

## Задания на лабораторные работы по использованию языка PHP

В данной лабораторной работе используется следующее программное обеспечение:

* web-сервер apache (<http://www.apache.org>);
* интепретатор языка PHP (<http://www.php.org>);
* СУБД mySQL (<http://www.mysql.com>);

**Внимание.** Любое полезное программное обеспечение можно найти по адресу <http://www.gnu.org>.

Необходимо разработать web-программу для линейного алгоритма. Ниже дан список вариантов заданий на лабораторную работу.

1. Даны два ненулевых числа. Найти их сумму, разность, произведение и частное.

2. Даны два числа. Найти среднее арифметическое их квадратов и среднее арифметическое их модулей.

3. Скорость лодки в стоячей воде V км/ч, скорость течения реки U км/ч (U < V). Время движения лодки по озеру T1 ч, а по реке (против течения) — T2 ч. Определить путь S, пройденный лодкой.

4. Скорость первого автомобиля V1 км/ч, второго — V2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили удаляются друг от друга.

5. Скорость первого автомобиля V1 км/ч, второго — V2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили первоначально движутся навстречу друг другу.

6. Найти периметр и площадь прямоугольного треугольника, если даны длины его катетов a и b.

7. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.

8. Найти длину окружности и площадь круга заданного радиуса R. В качестве значения Pi использовать 3.14.

9. Найти площадь кольца, внутренний радиус которого равен R1, а внешний радиус равен R2 (R1 < R2). В качестве значения Pi использовать 3.14.

10. Дана сторона равностороннего треугольника. Найти площадь этого треугольника и радиусы вписанной и описанной окружностей.

11. Дана длина окружности. Найти площадь круга, ограниченного этой окружностью. В качестве значения Pi использовать 3.14.

12. Дана площадь круга. Найти длину окружности, ограничивающей этот круг. В качестве значения Pi использовать 3.14.

13. Найти периметр и площадь равнобедренной трапеции с основаниями a и b (a > b) и углом alpha при большем основании (угол дан в радианах).

14. Найти периметр и площадь прямоугольной трапеции с основаниями a и b (a > b) и острым углом alpha (угол дан в радианах).

15. Найти расстояние между двумя точками с заданными координатами (x1, y1) и (x2, y2).

16. Даны координаты трех вершин треугольника (x1, y1), (x2, y2), (x3, y3). Найти его периметр и площадь.

17. Найти корни квадратного уравнения A·x2 + B·x + C = 0, заданного своими коэффициентами A, B, C (коэффициент A не равен 0), если известно, что дискриминант уравнения неотрицателен.

18. Найти решение системы уравнений вида A1·x + B1·y = C1, A2·x + B2·y = C2, заданной своими коэффициентами A1, B1, C1, A2, B2, C2, если известно, что данная система имеет единственное решение.

19. Дано целое четырехзначное число. Используя операции % и /, найти сумму его цифр.

20. Дано целое четырехзначное число. Используя операции % и /, найти произведение его цифр.