

Дмитрий Ченгаев

<http://www.bravoo.net>

Введение в PHP!

Базовые Знания Для Автоматизации Сайта



*Книга предназначена для свободного распространения в сети
Интернет.*

Запрещается коммерческое использование данной книги.

Ченгаев Дмитрий, © Copyright, 2007

Все права защищены.

Ченгаев Дмитрий – «Введение в PHP! Базовые знания для автоматизации сайта»

Издательство www.bravoo.net. 2007. – 43 с.

Электронный самоучитель «PHP! Как сделать современный веб-сайт!» предназначен исключительно для личного пользования и поставляется в виде как есть. Автор не несет никакой юридической, финансовой и иной ответственности за действия третьих сторон.

Воспроизведение всей электронной книги или любой ее части, форматирование или изменение структуры документа без письменного согласия автора Ченгаева Дмитрия Александровича запрещается.

В этой книге выражено исключительно только мнение автора по данному вопросу. Автор не несет никакой ответственности за действия, которые будут предприняты читателем, после прочтения электронной книги. Читатель принимает ответственность за использование всех материалов на себя.

Если Вы не согласны с данными условиями, можете удалить данную электронную книгу со своего компьютера!

Содержание:

ВВЕДЕНИЕ	4
ЧАСТЬ I. ВВЕДЕНИЕ В ВЕБ-ПРОГРАММИРОВАНИЕ	4
Глава 1. Что такое веб-программирование.....	4
Глава 2. PHP! Что это и почему именно он?	7
Глава 3. Что нужно для работы PHP?.....	9
Глава 4. Существуют ли PHP-редакторы и где их найти?	10
ЧАСТЬ II. ВВОДНЫЕ ЗАМЕЧАНИЯ ПЕРЕД РАБОТОЙ	14
Глава 1. Основы. Что, где и как.....	14
Глава 2. Немного об операторах.	15
Глава 3. Немного о командах языка.	16
Глава 4. Переменные.	17
Глава 5. Массивы.....	18
ЧАСТЬ III. НЕМНОГО ТЕОРИИ ПЕРЕД ПРАКТИКОЙ.....	20
Глава 1. Права доступа и изменение атрибутов файла.....	20
Глава 2. Базовые понятия передачи данных в веб-программировании.	22
Глава 3. А знаете ли вы, что такое Cookies?	25
ГЛАВА IV. ПИШЕМ ПРОГРАММЫ НА PHP САМИ.....	27
Глава 1. Формируем структуру сайта при помощи PHP.	27
Глава 2. Как узнать сколько раз скачали ваш файл?	29
Глава 3. Счетчик посещений.....	34
Глава 4. Проводим свое on-line голосование.....	36
ЗАКЛЮЧЕНИЕ	42

ВВЕДЕНИЕ



Здравствуйте, уважаемый читатель моей электронной книги. Для начала представляюсь, возможно, найдутся люди, которые еще со мной не знакомы. Мое имя Дмитрий Ченгаев. Я являюсь организатором и руководителем проекта <http://www.bravoo.net> – Техническая информация новичку для создания сайта. Эта электронная книга не единственная, на сайте также можно найти и другие мои публикации. Если Вам они интересны, то можете смело обращаться на мой сайт по ссылке приведенной выше.

Мне приходят письма от людей, которые создают свой сайт. Очень часто люди задают одни и те же вопросы. Эта книга, отвечает на цикл вопросов о языках веб-программирования. Здесь Вы сможете узнать о том, какие существуют языки веб-программирования, чем они отличаются, где их взять и т.д. Но главной темой все-таки является язык PHP. Почему для описания я выбрал именно его, вы поймете в соответствующей главе. В конце мы даже попробуем написать несколько своих собственных скриптов.

Хочу сразу предупредить, что эта книга в дальнейшем может переиздаваться, сюда может быть добавлен новый полезный материал и т.д. Кстати, это уже второе издание. Чтобы оставаться в курсе всех изменений и новостей настоятельно рекомендую Вам подписаться на мою бесплатную рассылку. Кроме того, в каждом выпуске этой рассылки публикуются самые свежие мои, и не только мои, статьи, посвященные техническим моментам создания сайта в сети Интернет. Я думаю, что эта информация могла быть полезной для Вас. Экономьте свое время.



Подписаться на рассылку можно по этой ссылке:

http://www.bravoo.net/viewpage.php?page_id=4

Вот собственно и все, что я хотел Вам сказать. Думаю, теперь настало самое время погрузиться в мир веб-программирования. Итак, приступим.

Часть I. Введение в веб-программирование

Глава 1. Что такое веб-программирование.

В последнее время, путешествуя по «Всемирной паутине», можно встретить все более и более интересные и в то же время сложные проекты. Теперь уже не модно иметь сайт, который бы просто хранил информацию. Современные сайты проводят опросы, что-нибудь вычисляют, выдают определенные формы для заполнения, в общем, взаимодействуют с пользователем. А чтобы все это делать, без программирования не обойтись. Вот и выделилась в сети отдельная область - web-

программирование. Знакомо ли вам это слово? Может быть вы, что-нибудь даже и слышали по этому поводу? Сейчас в Интернет и не только в нем, можно найти огромное количество информации по этому вопросу. Можете зайти на любой форум, посвященный строительству сайта, и вы увидите, как популярна сегодня тема веб-программирования.

Когда я искал ответы на вопросы по этой теме, то можно сказать «перелопатил» море информации. Иногда, прочитав даже умную книгу по веб-программированию, опыта и знаний совсем не прибавляется. Все написано слишком умным языком, много теории и очень мало практики. Все знания появляются отрывками, а в целом никакого представления не формируется. Знакома ли вам такая ситуация? Но данная книга не относится к таковым. Это скорее всего руководство к действию, чем изложение теории. А начнем мы с вами с того, что представляет собой веб-программирование на сегодняшний день.

Итак, само слово веб-программирование уже говорит само за себя – это программирование, которое предназначено для работы в Интернет. Вообще, Интернет, в своей основе, это сеть. Чтобы вся информация успешно передавалась по проводам, здесь действуют свои правила и законы. Самая главная особенность Интернет – это ограниченная скорость передачи информации. Передавать видео, аудио или другой объемный формат данных, здесь достаточно затруднительно. Именно поэтому Интернет имеет свою собственную компьютерную специфику и язык программирования для него тоже особенный.

Если взглянуть на все языки веб-программирования, то можно выделить две группы: клиентские и серверные. Перед тем, как приступить к их рассмотрению, немного разберемся с определениями. Ведь мы же должны иметь представление, о чем собираемся говорить? Определений будет не много, всего два.

Сервер – это тот компьютер, на котором храниться сайт. Еще под этим термином понимают ту программу, которая обрабатывает запросы браузера пользователя к любому документу.

Клиент. Проще говоря, это сам пользователь, а вернее его браузер. Когда пользователь Интернета обращается с запросом к серверу, то он является клиентом.

Клиентские языки.

Начнем с того, что эти языки выполняются на компьютере пользователя. Если говорить более точно, то их выполняет сам браузер. Обычно клиентские языки встраиваются в html-код страницы. Таким образом, чтобы их увидеть, достаточно открыть web страницу с помощью текстового редактора.

Так как клиентские языки выполняются на компьютере пользователя, то здесь только пользователь решает, нужны они ему или нет. Он может просто их отключить и браузер не будет ничего выполнять. Также языки могут не работать на старом браузере, он просто может их не поддерживать.

Одним из недостатков клиентских языков является не умение отправлять файлы на сервер. Это значительно ограничивает их применение. Например, о гостевых книгах можно вообще забыть.

Кроме того, раз любой пользователь может очень легко просмотреть код программы, то о безопасности можно забыть тоже. Если вы захотите закрыть доступ

к некоторым файлам, с помощью таких языков, это вряд ли удастся. Достаточно просто просмотреть HTML – код страницы, чтобы узнать все пароли.

Однако, раз клиентские языки добились такой популярности, то у них должны быть и преимущества. Да, преимущества есть и еще какие!

1) Они не отправляют данные на сервер, что делает их работу значительно быстрее.

2) Не требуется никакого дополнительного программного обеспечения, все необходимое есть в браузере клиента.

3) За пользование серверными языками компания, предоставляющая хостинг, требует денег, а клиентскими нет.

Самыми распространенными клиентскими языками являются: JavaScript и VisualBasicScript (VBS). Для того, чтобы браузер мог их понимать и выполнять в него встроен специальный инструмент – интерпретатор. JavaScript, разработан компанией Netscape и первоначально использовался только для браузера Netscape Navigator. В настоящее время этот язык получил очень большую популярность. VisualBasicScript (VBS) это аналог клиентского языка от компании Microsoft.

Как уже говорилось ранее, чтобы эти языки начали работать, нужен браузер, который бы их поддерживал. Я бы посоветовал использовать следующие браузеры:

1) Internet Explorer (www.microsoft.com)

2) Opera (www.opera.com)

3) Mozilla (www.mozilla.org)

Серверные языки.

Серверные языки предназначены для выполнения непосредственно на самом сервере, специальной программой. Это значит, что пользователь не сможет запустить их с помощью одного только браузера. Обычный пользователь вообще не может видеть код и сами программы, написанные и выполняемые на серверном языке. Он видит только результат их действия.

Серверные языки предоставляют веб-программисту больше возможностей, чем клиентские. Используя их можно обмениваться данными с сервером, чего мы были лишены при использовании клиентских языков.

Наибольшую популярность, среди серверных языков получили:



- PHP (PHP: Hypertext Preprocessor) (www.php.net).



- Perl (www.perl.com)

Т.к. серверные языки могут взаимодействовать с удаленным компьютером (сервером), возникает вопрос, а где же мы будем хранить результаты этого взаимодействия? Например, гостевая книга предполагает сохранение на сервере информации об имени пользователя и данные оставленного им сообщения. Если, данных у вас не слишком много, можно конечно хранить их в обычном текстовом файле. Но по мере роста любого сайта, приходится хранить огромное количество информации. Именно для этой цели существует База Данных (БД). Это своего рода библиотека, где вся информация аккуратно разложена по полочкам. Но просто одной БД еще не достаточно. Оказывается, когда информации становится очень много, найти то, что нам необходимо становится затруднительным, не говоря уже, что все необходимо обработать. Почему бы не поручить это компьютеру? Так и поступили. В настоящее время БД успешно обрабатываются компьютером и системы, которые за все это отвечают, называются Системами Управления Базами Данных (СУБД). В них можно хранить всю необходимую вам информацию, даже пароли (причем в зашифрованном виде). Взаимодействие пользователя с БД обеспечивает специальный язык запросов - Structured Query Language (SQL) (Язык структурированных запросов).



Самым известным СУБД является MySQL (www.mysql.com).

Глава 2. PHP! Что это и почему именно он?

По моему мнению, PHP является одним из самых легких для понимания языков программирования.

PHP имеет синтаксис, аналогичный синтаксису на языке программирования С. Если вы знакомы с С, то вам будет значительно легче разобраться и с PHP. Но если такого опыта нет, то не расстраивайтесь, будем учиться.

Т.к. PHP выполняется на стороне сервера, выполнение его команд может очень сильно загружать отдаленный компьютер. Именно поэтому, компании, с бесплатным хостингом так неохотно предоставляют PHP. Хотя такие компании есть и даже в Рунете. Попробуйте обратиться сюда:

- <http://www.hl.ru>
- <http://webservis.ru>
- <http://wallst.ru>
- <http://www.hut.ru>

Но если вы собираетесь серьезно заниматься веб-программированием, без платного хостинга не обойтись. Только там можно найти нормальный сервис, но за него нужно платить. Бесплатный сыр только в мышеловке.

PHP специально предназначен для работы в сети Интернет. Сегодня, если Интернет провайдер не предоставляет возможность использовать PHP, то с ним никто не будет иметь дела.

PHP расшифровывается как "PHP: Hypertext Preprocessor" (т.е. гипертекстовый процессор предварительной обработки). Давайте сразу определимся: т.к. PHP относится к серверным языкам, то он может выполняться только на том компьютере, на котором есть специальное программное обеспечение. Т.е. если Вы просто так запустите файл с PHP кодом на Вашем компьютере, ничего не произойдет. Код не будет работать. Вам нужен домашний сервер, т.е. специальная программа для работы с такими технологиями. Кстати наличие сервера также необходимо для технологий SSI и CGI. На сегодняшний день одним из лучших домашних серверов является Apache. Но сейчас мы не будем подробно останавливаться на этом вопросе и рассмотрим его более подробно в следующем разделе.

Когда я создавал свой сайт, мне нужно было выбрать хостинг. Я никак не мог определиться, нужен ли мне PHP или нет. Дело в том, что за сайт без его поддержки платить придется значительно меньше. Чтобы разобраться в этом вопросе, нужно понять, что же PHP умеет делать. Давайте этим и займемся.

Любое программирование, какое оно бы не было, предназначено для автоматизации. Не исключение и web- программирование. Для того, чтобы просто создать свой сайт не обязательно знать и понимать, что такое PHP. Вполне достаточно будет знаний HTML. У вас будет нормально функционирующий сайт. Только возможностей у него будет маловато. Давайте сейчас разберемся, что же это за возможности.

1) По мере того, как web-проект начинает разрастаться управлять им вручную становится практически невозможным. На помощь приходит веб-программирование и автоматизация.

2) Чтобы сайту выдержать жесткую конкуренцию он обязан быть интерактивным (т.е. взаимодействовать с пользователем). Необходимостью уже стали различные формы, опросы и др. которые пользователи должны заполнять. Согласитесь, что данные, которые они заполнили нужно как-то обработать. Этим и занимается PHP.

3) В последнее время на сайтах стала появляться интересная возможность – гостевая книга. Т.е. посетители заходят на web-страницу, прямо в on-line оставляют свой отзыв, который затем будет виден другим посетителям, зашедшим на эту страницу.

4) С помощью PHP можно сделать мини Интернет магазин. Принцип его действия будет достаточно простой. На Ваш сайт устанавливается определенная форма. В поля этой формы покупатели будут вводить необходимую информацию о себе и о приобретаемом товаре. В итоге сформируется определенная форма заказа. Далее она отправляется по электронной почте на Ваш e-mail. Самое важное здесь в том, что покупателю не нужно даже пользоваться электронной почтой.

5) Для покупателей Интернет магазина было бы очень удобно, в случае, если они покупали бы сразу несколько товаров, и цена подсчитывалась бы автоматически. Т.е. web-страница выполняла бы роль некоего калькулятора.

6) PHP обеспечивает очень интересную возможность, авторизации доступа. Допустим, у Вас есть какие-нибудь ресурсы, доступ к которым должен быть выделен только определенным людям. Видели формы на сайтах с двумя разделами: Логин: ***** и Пароль: *****? Это и есть та самая авторизация.

7) Для создателя сайта очень важно знать какое мнение, у его посетителей, о товарах расположенных на сайте. Для этого существует специальная система оценки. Т.е. посетитель может оценить товар, например по пятибалльной шкале: отлично, хорошо, удовлетворительно, ...

8) Как Вы думаете, а на каком принципе работают счетчики посещений и скачиваний которые используются на web-сайтах? Все это PHP.

9) И напоследок, код PHP может быть встроен непосредственно в html документ. Это, несомненно, выгодно отличает его от Perl.

Ну, что, заинтересовались? Некоторые из описанных возможностей, вы сможете установить на свой сайт, уже после прочтения данной книги. Но если все это вам не нужно, можете забыть про веб-программирование и удалить со своего компьютера данную книгу, она вам не понадобится.

Надеюсь, вы приняли свое решение, тогда начнем. Для начала рассмотрим те места, где можно более подробно узнать о PHP.

Первое, на что следует обратить внимание является официальный сайт разработчиков www.php.net. На этом сайте есть много полезной информации и дистрибутивов, но, к сожалению, здесь все на английском.

Но не расстраивайтесь, в Рунете, тоже есть свои места.

<http://php.spb.ru>. Там вы можете найти документацию, необходимые дистрибутивы, статьи, форум и многое другое. Но самое главное, здесь все на русском языке.

Также, могу посоветовать

<http://www.phpclub.net>. Этот сайт – клуб PHP разработчиков.

Я уверен, что, пользуясь данными сайтами, вы сможете без проблем найти и остальные необходимые ресурсы.

Все возможности, которые здесь описаны – это только самое основное, что может PHP. Итак, если вы хотите, чтобы ваш сайт был популярен, современен и моден, теперь вы знаете, в каком направлении нужно двигаться. А я лишь буду пытаться направлять вас в нужном направлении.

Глава 3. Что нужно для работы PHP?

Еще раз напомним, что PHP – это серверный язык. Для его работы недостаточно просто написать код и запустить его в браузере (как, например, JavaScript). Ничего

работать не будет. Нужна специальная программа, которая бы работала с PHP. Конечно, можно проверить этот код на Интернет сервере, который поддерживает PHP. Но только представьте себе, что каждый раз нужно будет выходить в Интернет, чтобы проверять хотя бы одну страничку. Может быть, существует более рациональное решение? А можно ли тестировать PHP файлы прямо на домашнем компьютере?

Да, такое решение есть, для этого нужно установить виртуальный web-сервер, который смог бы работать с технологией PHP. На сегодняшний день одним из самых лучших web-серверов является Apache. Итак, где его можно найти?

Для начала можно обратиться к официальным ресурсам.

Официальный сайт Apache: <http://www.apache.org>

Официальный сайт PHP: <http://www.php.net>

Сразу скажу, что качать придется много: около 10 Мб.

Если скачать такой объем информации не представляется возможным, могу порекомендовать урезанную версию http://php.spb.ru/php/install_module.html. Там же есть инструкции, очень важно уделить им достаточно внимания. Иначе Вы рискуете тем, что у вас ничего не будет работать.

Ну и напоследок посмотрите на:

Всероссийский Клуб Веб-мастеров: <http://www.webclub.ru>.

Клуб разработчиков PHP: <http://www.phpclub.net>.

Лаборатория dk: <http://www.dklab.ru>.

На этом закончим данную главу. Извините, но я считаю излишним подробно описывать процесс установки веб-сервера на ваш компьютер. По ссылкам, которые приведены выше, можно найти достаточно информации по данному вопросу. Я же, своей задачей ставлю направить вас в нужном направлении. В крайнем случае, если у вас не будет ничего получаться, напишите, и мы попробуем разобраться вместе. info@bravoo.net

Глава 4. Существуют ли PHP-редакторы и где их найти?

Прежде чем начать эту главу, мне бы хотелось сделать маленькое введение в свою историю. Мне кажется, это было полезно многим читателям. Может быть, кто-нибудь увидит в моих словах проблемы, которые сейчас волнуют и их. Если это действительно так, значит вы попали на нужные страницы и мне остается только поздравить вас. Итак, начнем.

Когда-то я начинал создавать свой сайт, изучил html и начал писать код в обычном блокноте. Прошло некоторое время, и мне показалось, что как-то это совсем не продуктивно писать столько текста, неужели нельзя как-нибудь все автоматизировать? Тем более, когда я просматривал код профессиональных веб-страниц, там было столько текста, что меня бросало в ужас. «Неужели они это все набирают? Какая же тогда должна быть скорость печати и сколько на это нужно потратить времени?»: вот такие вопросы возникали в моей голове. Я начал искать

на них ответы. Искать пришлось совсем не долго. Оказывается, существуют html-редакторы. Потратив еще немного времени на поиски, я установил на свой компьютер FrontPage. Удивлению не было предела! Теперь на создание страницы уходило в десятки раз меньше времени!

Все бы было хорошо, но через некоторое время я узнал, что существуют языки веб-программирования, и мне захотелось улучшить с их помощью свой сайт. Выбор упал на язык PHP. Сначала, пока программы были совсем маленькие, проблем не возникало. Можно было с легкостью пользоваться html-редактором и писать коды. Но позднее программы становились все более массивными, и возникла точно такая же проблема, которая возникала у меня с html-кодом: хотелось бы как-нибудь все автоматизировать. Мне был нужен PHP-редактор.

Именно вопросу PHP-редактора и посвящена эта глава. Я надеюсь, что после ее прочтения вы раз и навсегда забудете, что когда-то у вас была такая проблема.

Уважаемый читатель, если вы задумали серьезно заниматься веб-программированием на языке PHP, то вам просто необходим PHP-редактор. Наверняка, у вас уже возникал вопрос, а существует ли такие редакторы вообще? На самом деле было бы даже удивительно, если бы их не было. Конечно, они существуют. Сейчас я постараюсь сделать краткий обзор PHP-редакторов. Вам остается только выбрать и установить наиболее подходящий редактор для вас.

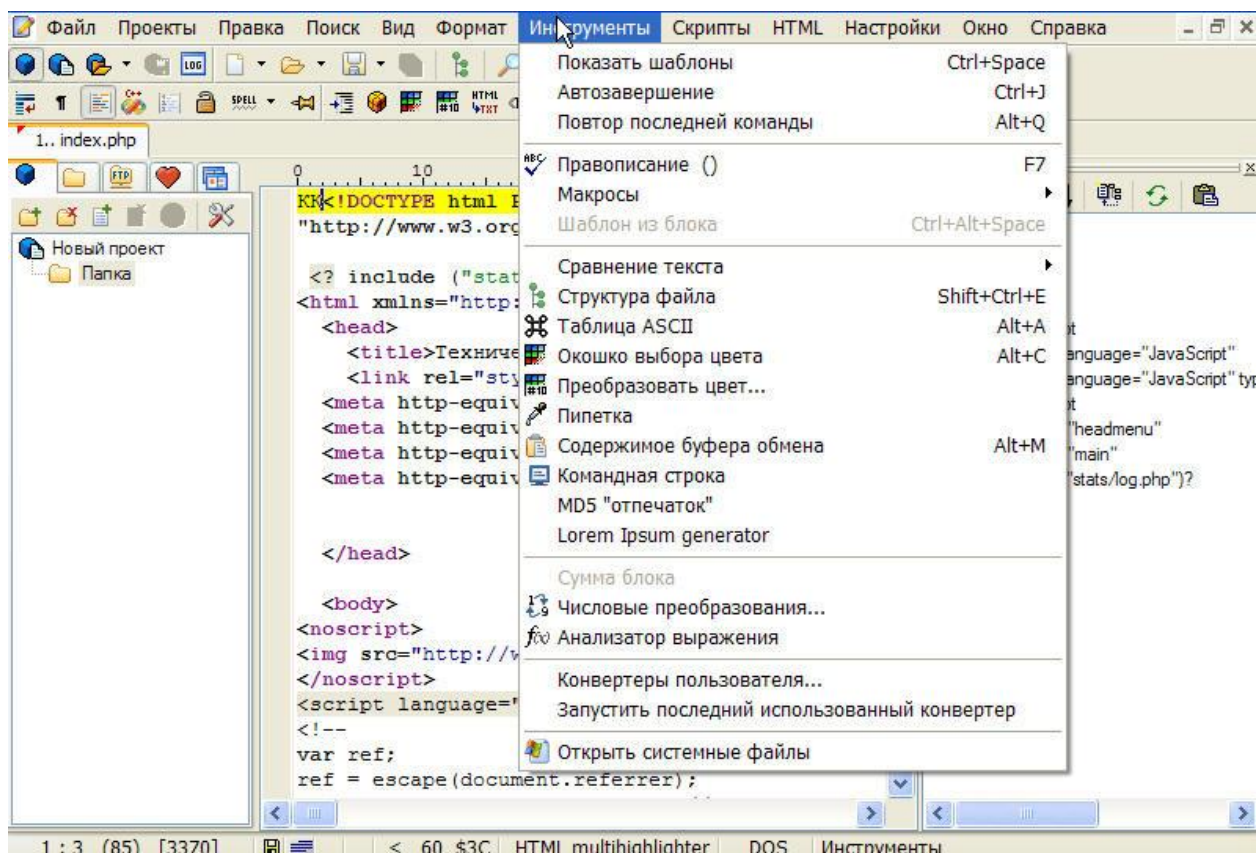
Zend Studio 5.5

Начнем, пожалуй, с самого навороченного, на мой взгляд, редактора Zend Studio 5.5 (Zend Development Environment). Здесь есть все необходимые функции, переменные, классы, отличная графика, там, где надо подсветит... в общем все, что душе угодно.

Цена этого продукта 299\$, но в сети можно найти crack-и. Единственная проблема – это скачать. Весит он около 70 Мб. Вот ссылка для скачивания http://www.zend.com/products/zend_studio.

PSPad версии 4.5.1

Мне очень понравился этот редактор. Все очень красиво и русифицировано.



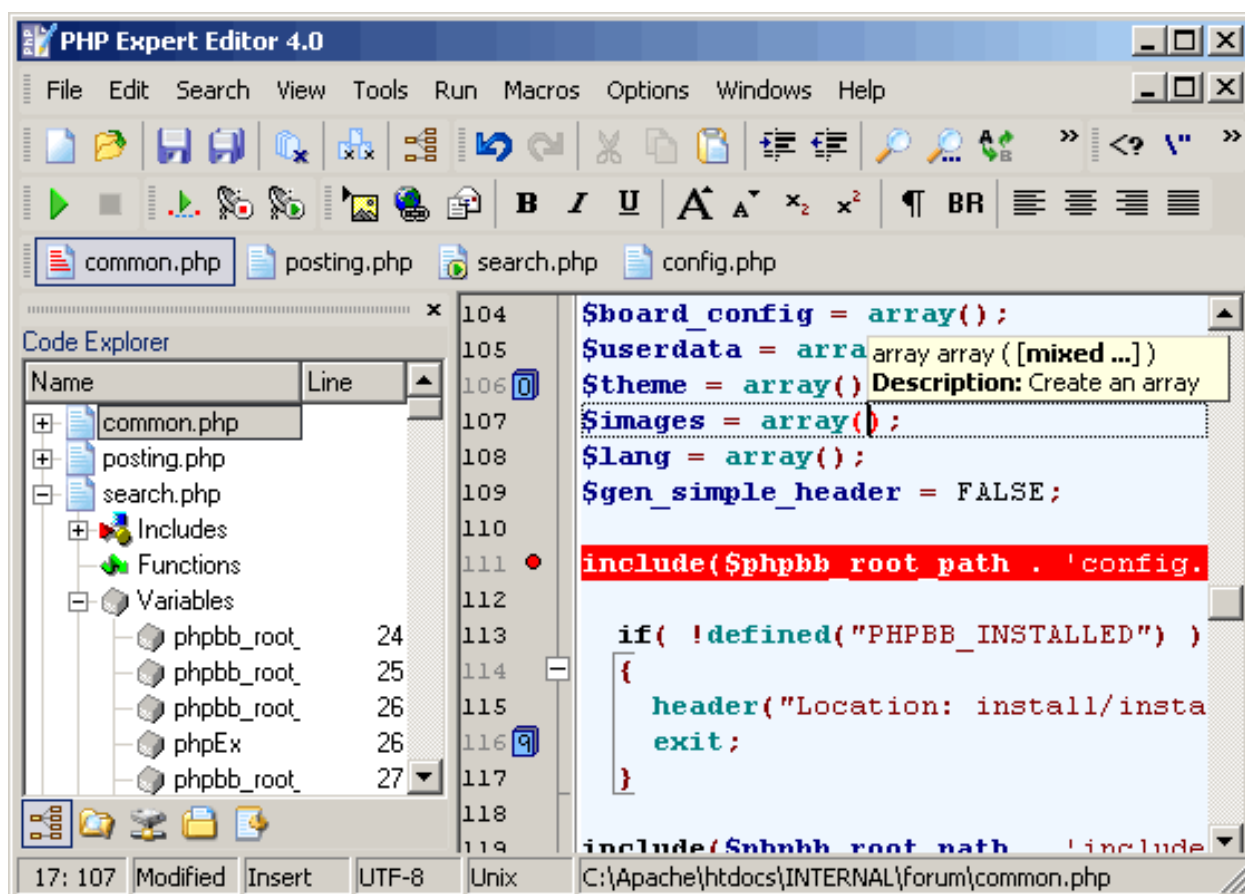
Скачивать нужно около 3 Мб.

Сайт поддержки <http://www.pspad.com/ru/whatsnew.htm>

Вот ссылка на саму программу: <http://pspad.cincura.net/files/pspad451en.cab>

PHP Expert Editor

Еще один навороченный редактор Версия 4.0 (30-03-2006). Разработана это программка специально для PHP-мастеров. Одно из его главных преимуществ – это то, что он имеет встроенный HTTP сервер и отладчик PHP скриптов. Это значит, что вам не нужно каждый раз выходить в Интернет, чтобы проверить работоспособность скрипта. Кроме того, здесь есть встроенный браузер, ftp-клиент, проверка синтаксиса и многое другое.



Скачать программу можно по ссылке: http://www.ankord.com/download/phpxedit_40b.zip. Занимает программа всего 3,6 Мб. Для жителей России она бесплатна.

Вот самые основные редакторы, с которыми мне приходилось иметь дело. Кроме того, редактировать PHP можно с помощью html-редактора Dreamweaver. Но за универсальность приходится платить функциональностью.

Есть еще много менее известных программ, я не вижу смысла перечислять их все. При желании можно все найти при помощи поисковика. Но вот некоторые из них:

- EditPlus2
- SciTE
- DinoEditor 1.5
- Macromedia HomeSite 4.5
- PHP DESIGNER 2006 Beta
- PHP Edit
- и другие.

Выбирать только вам. Надеюсь я помог вам сделать правильный выбор с наименьшими затратами времени и денег.

Часть II. Вводные замечания перед работой

Глава 1. Основы. Что, где и как.

Прежде, чем начать изучать PHP, давайте разберемся, как он работает. PHP выполняется на сервере. Представим себе, что на удаленном сервере находится PHP-страница. Как браузер ее обрабатывает? Он сначала посылает серверу запрос на страницу с PHP-кодом. Сервер отдает эту страницу специальной программе, которая умеет распознавать и работать с PHP (интерпретатору). Интерпретатор выполняет все необходимые действия, создает новый HTML-код и посылает его обратно серверу, а тот в свое время посылает обработанную страницу клиенту. Таким образом, PHP – код никак не попадает в браузер, а это значит, что клиенту увидеть его нельзя.

Вообще, для написания PHP достаточно любого текстового редактора (например, блокнота). Хотя и существуют специальные программы-редакторы, но использовать их совсем даже не обязательно. Где будете работать вы, зависит только от вашего вкуса.

Создать PHP-документ достаточно просто. Его код просто вставляется в HTML-страницу. Это похоже на вставку сценариев JavaScript и VBScript, только синтаксис другой. Также не забудьте присвоить вашему файлу соответствующее расширение (*.php). Итак, для того, чтобы дать интерпретатору понять, что далее будет идти php-код нужно соответствующим образом его выделить. Делается это с помощью угловых скобок, знака вопроса и указания языка. Эти знаки называются дескрипторы.

```
<?php
...
Команда 1;
Команда 2;
...
?>
```

Однако, использовать слово «php» в дескрипторе не обязательно. Можно обойтись и без него:

```
<?
...
Команда 1;
```

Команда 2;

...

?>

Это немного сокращает код, но, чтобы интерпретатор понимал такую форму записи, его нужно настроить. Таким образом, вы рискуете, что на некоторых серверах ваш код работать не будет. Поэтому я все-таки советую использовать полную запись.

Заметьте, что после каждой команды ставиться знак «;». Так PHP отличает одну команду от другой. Однако, в некоторых случаях «;» не ставят:

- после условных операторов (if, switch)
- после операторов цикла (for, while и др.)
- также после последней в программе команды.

Иногда необходимо, пояснить, какой-то момент кода. Для этого используют комментарии – фрагменты текста, которые не воспринимаются интерпретатором как PHP-код. Чтобы вставить комментарии, нужно на соответствующей строке нужно написать.

Команда; //Здесь будет комментарий.

Таким образом, комментарии создаются конструкцией «//».

Вот, собственно, все основные тонкости языка. Конечно, если раньше вы никогда не занимались программированием, то будет тяжело. Но ведь нужно же когда-то начинать.

Глава 2. Немного об операторах.

Никакой язык программирования не может обойтись без математических операций. В PHP тоже встроены операторы, которые позволяют производить математические вычисления. На самом деле этих операторов много, я лишь приведу только те, которые могут встретиться вам в ходе чтения книги.

Основные арифметические операции:

$a + b$ — сложение

$a - b$ — вычитание

$a * b$ — умножение

a / b — деление

$a \% b$ — остаток от деления a на b

`$переменная++`; - запись эквивалентна следующей - `$переменная = $переменная + 1`. Значение `$переменной` увеличивается на 1.

`$переменная--`; Аналогично. Значение `$переменной` уменьшается на 1.

Кроме математических операций, нам придется еще использовать различные циклы и условия. А какое условие без логических операций? В PHP есть определенные правила написания этих условий. Новичкам обычно бывает тяжело привыкнуть именно к таким правилам написания.

Основные логические операции:

`«==»` - знак равенства. Обратите внимание, что именно два знака равенства! Один знак `«=»` является оператором присваивания (`Переменная = значение`).

`«>»`, `«<»`, `«>=»`, `«<=»` - больше, меньше, больше или равно, меньше или равно.

`«!=»` или `«<>»` - знак неравенства.

`a и b` - условия

`a && b` – действие будет выполнено, если выполняется И условие `a`, И условие `b`.

`a || b` - действие будет выполнено, если выполняется ИЛИ условие `a`, ИЛИ условие `b`, ИЛИ оба одновременно.

Глава 3. Немного о командах языка.

В PHP, как и в любом другом языке программирования есть команды, с помощью, которых мы можем выполнять некоторые действия. Этих команд достаточно много. Чтобы перечислить их все, несколькими страницами не обойтись. Я же хочу рассказать о тех командах, которые мы будем использовать для написания наших программ. Это только самое основное, что касается других команд, мы будем разбирать их по ходу рассмотрения программы.

Итак, начнем:

1) **echo** "текст". Это одна из самых часто используемых команд. Ее можно встретить практически в любой php-программе. Она выводит на web-страницу какой-либо текст, заключенный в кавычки.

2) **include** "имя файла". Эта команда вставляет содержимое одного файла в другой. Файл именно вставляется, т.е. пути к картинкам уже нужно указывать от того файла, где вставляется команда **include**.

Также нам встретятся циклы и условия:

1) **if** (условие) {...команды, которые должны выполняться, если условие верно...;} **else** {...команды, которые должны выполняться, если условие неверно...}

В условии могут использоваться соответствующие логические операторы, которые описаны выше.

Можно использовать несколько условий, используя знаки «&&» - И «||» - ИЛИ. Тогда нужно писать так:

```
if (условие 1 && условие 2)
```

2) **for** (начальное значение счетчика, условие продолжения цикла, изменение счетчика на каждом цикле) { . . . команды. . . ; }

```
while (условие) { . . .команды. . . }
```

Все! Больше не буду загружать вас скучными определениями. Остальное рассмотрим на практике. Я считаю, что так будет более продуктивно.

Глава 4. Переменные.

Переменные – это контейнеры, в которых содержатся данные. Эти данные, при необходимости, можно из контейнеров брать. Переменные вы встретите практически в любой программе.

Для создания переменной нужно дать ей имя и значение. Чтобы PHP мог отличать переменные от обычного текста или команды, их имя должно начинаться со знака доллара («\$»). Например, \$a – объявление переменной a. Также существуют определенные ограничения на имя переменной:

- оно не должно содержать пробелов
- не должно содержать знаков апострофа и некоторых других символов.

Имя переменной иначе называют идентификатором. Желательно, делать его осмысленным. Потом легче ориентироваться. Например, переменную текущего года лучше назвать \$year = 2006 (а не \$var1=2006).

Чтобы создать переменную, нужно написать ее имя, а затем, через знак равенства (присваивания) значение.

```
$year = 2006
```

```
$name = "Вася" (Текстовая информация присваивается в кавычках)
```

Переменным обычно присваивают следующие данные:

- Числа (1;50;4534)
- Текст (т.е. набор букв) (Коля, Вася, ...)
- Массив (совокупность данных под одним именем)

Чтобы лучше понять, для чего нужны, переменные рассмотрим простой пример:

```
<?php
```

```
$name = "Вася";
```

```
echo "Привет " $name;
```

?>

После выполнения этой программы мы увидим веб-страницу с текстом «Привет Вася».

Таким образом, один раз объявив переменную в дальнейшем на нее можно просто ссылаться.

Глава 5. Массивы.

Еще один термин, с которым нам необходимо познакомиться – это массив. Массив – это набор данных, объединенных под одним именем. Храниться он может в переменной. Например: переменная `$ucheniki` может содержать в себе информацию обо всех учениках, которые учатся в определенном классе (Сидоров, Петров, Иванов,...). Когда располагаешь информацией таким образом, ее очень легко обрабатывать.

Рассмотрим несколько способов создания массива.

Одним из самых легких способов создания массива является следующая форма записи:

```
$имя массива[номер элемента] = значение
```

Номер элемента также называют ключом (key) и индексом (index).

```
$ucheniki[0] = «Иванов»;
```

Это означает, что нулевой элемент массива будет содержать текстовую информацию «Иванов». Да, вы не ослышались, нулевой. Дело в том, что в PHP нумерация начинается с нуля. Продолжаем:

```
$ucheniki[1] = «Петров»;
```

Теперь мы задали первый элемент. Продолжая действовать таким образом, можно создать массив.

Еще одним способом задания массива является:

```
$ucheniki = array("Иванов","Петров","Сидоров");
```

Здесь слово «array» обозначает «массив». А, если нам необходимо, чтобы отсчет начинался с 50 (или другой цифры), а не с нуля, нужно написать:

```
$ucheniki = array(50 => "Иванов", "Петров", "Сидоров");
```

Теперь, когда создавать массивы мы научились, возникает другая проблема: как выводить информацию с массива пользователю? Для просмотра элементов массива существуют две функции: `var_dump()` и `print_r()`. Допустим, нам нужно вывести значение элементов массива `$ucheniki`, который мы заполнили ранее. Это можно сделать с помощью двух функций:

```
1) print_r($ucheniki);
```

В результате получим:

```
Array
```

```
(
```

```
[0] => Иванов
```

```
[1] => Петров
```

```
)
```

```
2) var_dump($ycheniki);
```

В результате получим:

```
Array(2)
```

```
(
```

```
[0] => "Иванов"
```

```
String(6)
```

```
[1] => "Петров"
```

```
String(6)
```

```
)
```

Как видите, более полную информацию, дает второй способ.

Для работы с массивами используют разные функции. Одну из них мы сейчас рассмотрим. Бывают ситуации, когда необходимо сообщить программе, сколько всего элементов в нашем массиве. Конечно, когда вы знаете это число, то можно просто указать его конкретной цифрой (например, 5), но это не всегда получается. Дело в том, что количество элементов может изменяться, также массивы могут формироваться автоматически и программист просто не знает, сколько всего получилось элементов. Специально для таких ситуаций создана специальная функция `count($имя_массива)`.

```
$n = count($имя_массива);
```

В этом случае переменной `n` будет присвоено значение количества элементов в массиве.

Часть III. Немного теории перед практикой.

Глава 1. Права доступа и изменение атрибутов файла.

Прежде, чем мы приступим к практическим заданиям, мне бы хотелось немного остановиться на теоретической части и разобрать несколько очень важных тем.

Одной из самых основных проблем, с которой сталкиваются новички, это установка прав доступа на файл. На этой теме мне бы хотелось остановиться в первую очередь. Если вы когда-нибудь пробовали установить какой-нибудь скрипт на ваш сайт, то наверняка видели требование, в readme файле: «У файла такого-то выставить права `chmod 755`».

Например, у меня сразу возник ряд вопросов: «что значит выставить права?» и «Что это за `chmod 755`?».

Дело в том, что когда скрипт пытается записать какие-то данные в файл, он должен получить соответствующее разрешение от операционной системы. Чтобы дать это разрешение, на файл должны быть установлены соответствующие права (атрибуты).

Права доступа (CHMOD) – это атрибут, которым снабжен каждый файл и каждая папка на веб-сервере. Благодаря этим правам, можно указывать какие действия разрешено производить с ними.

Если не установить эти права, то скрипт может вообще не работать, хотя весь код будет правильным.

Начнем с теории. Владелец сайта имеет право разрешать или запрещать чтение, запись или выполнение (если это программа) файлов для посетителей своего сайта.

Для указания прав доступа к файлам используются следующие символы:

r – Пользователи могут читать файл.

w – У пользователей появляется право изменять содержимое файла и копировать его содержимое.

x – разрешает запускать файл на выполнение.

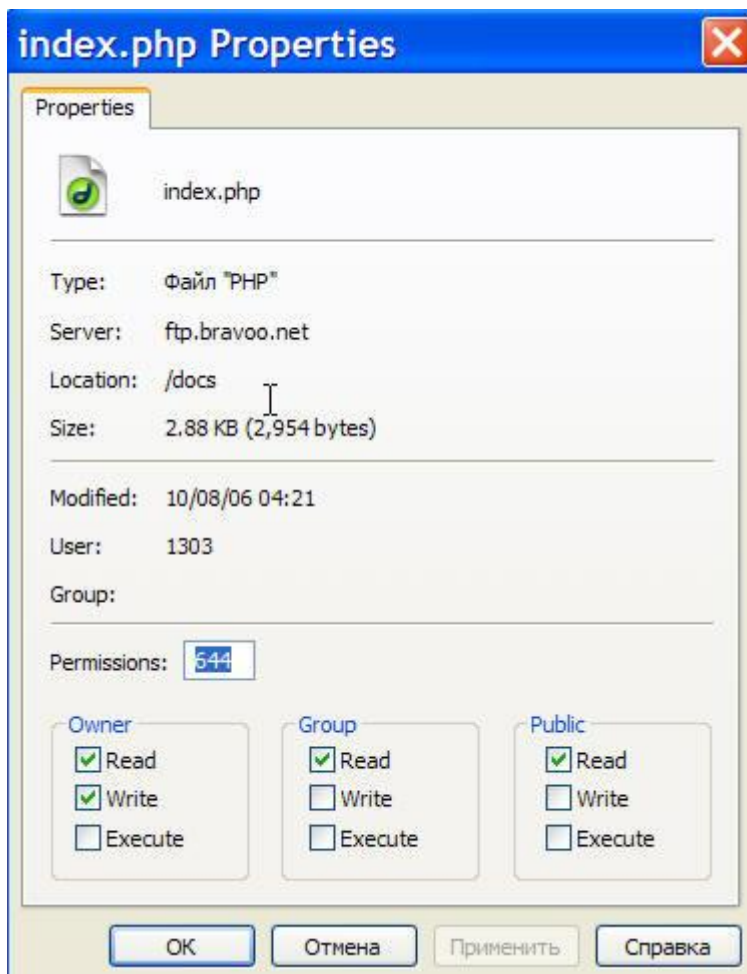
Для указания тех пользователей, которые имеют соответствующие права, используются символы:

u – владелец файла (User, Owner)

g – тот, кто принадлежит к той же группе, что и владелец сайта. (Group)

o – все пользователи (Public)

Для установки прав доступа можно воспользоваться FTP-клиентом. Например, я пользуюсь CuteFTP. Для этого нужно нажать на нужном файле правой кнопкой мышки и выбрать пункт Properties/CHMOD.



После этого появиться окошко со свойствами нашего файла или папки, где нужно поставить галочки в соответствующих разделах. Либо выставить конкретное численное значение прав, в текстовое поле Permissions (Например: 666, 777 или др.). После этого нажимаем ОК и права будут установлены.

Вы наверняка уже заинтересовались, что это за цифровые обозначения (666 или 777). На самом деле все очень просто. Таким образом распределяются права для соответствующих групп.

4 – это право на чтение (read)

2 – право на запись (write)

1 – право на выполнение (обычно используется для программ) (execute)

Так вот, в номере 666, первая цифра обозначает права для группы Owner (владельца сайта т.е. для вас), вторая для Group (для людей работающими вместе с вами в одной группе), и, наконец, третья для всех остальных – Public.

В нашем случае 6 получается путем сложения чисел $4+2=6$. Т.е. файл будет доступен для чтения и записи.

Вот цифры, которые могут еще встретиться:

3 (2+1) – разрешена запись и выполнение файла.

5 (4+1) – чтение и выполнение.

7 (4+2+1) – разрешены все доступные операции.

Все, больше вариантов нет.

Как видите ничего сложного. Кроме программы CuteFTP, можно также пользоваться:

- Windows Commander (Total Commander). Через главное меню «Файл-Изменить атрибуты».
- FAR Manager. С помощью нажатия сочетания клавиш Ctrl + A.
- И др.

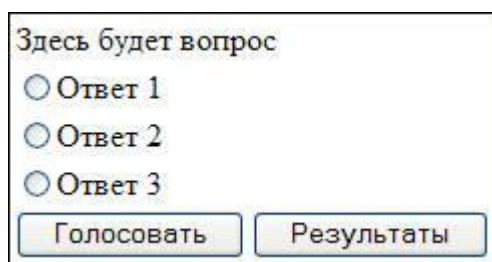
Надеюсь теперь вам понятно, зачем и как нужно устанавливать атрибуты файла. Прошу вас хорошенько запомнить, как это делать. Дело в том, что для дальнейшего изучения книги, эти знания очень даже пригодятся. Читайте дальше и вы в этом убедитесь.

Глава 2. Базовые понятия передачи данных в веб-программировании.

Следующий очень важный теоретический вопрос звучит так: «Каким образом передаются данные в веб-программировании?».

Очень важно понимать, как это на самом деле происходит, потому что данные передаются практически в любом скрипте, и столкнуться вам с этим все равно придется.

Возьмем хотя бы программу on-line голосования, пользователю предлагается ознакомиться с вопросом и выбрать подходящий вариант ответа.



После того, как нужный ответ будет отмечен и пользователь нажмет на кнопку «Голосовать» данные его голосования должны будут передаться программе на языке веб-программирования. О том, как они будут передаваться мы, и поговорим в данной главе.

Далее я буду использовать один термин, чтобы между нами не было непонятных моментов, я бы хотел остановиться на нем подробнее и разобраться что это такое.

HTML – форма, знакомо ли Вам такое определение? Вы, наверняка, ее уже видели и даже имеете представление, что это такое. Не верите? Тогда взгляните на следующие иллюстрации и вспомните, видели ли вы нечто подобное.



Отдаете:
WMR ▼

Получаете:
WMZ ▼

Выбрал!

Если бы завтра были выборы президента РФ, кому бы Вы отдали свой голос?

- ☐ Никому (против всех)
- ☐ Путину
- ☐ Медведеву
- ☐ Иванову

РАССЫЛКА НОВОСТЕЙ

Ваш E-mail

Ok

Параметры рассылки

То, что вы видите сейчас на иллюстрациях и есть те самые формы. Это специальные области HTML-документа, которые содержат средства взаимодействия с пользователем: кнопки, текстовые поля, радио-кнопки (переключатели), списки, выпадающие списки и др.

Очень часто бывают такие ситуации, когда нужно спросить у пользователя какую-нибудь информацию. Например, e-mail адрес для того, чтобы подписать его на почтовую рассылку, узнать более приемлемый для пользователя способ оплаты (чтобы он выбрал его из списка) и многое другое. Такую функцию и выполняют формы, они «спрашивают» у пользователя его мнение и заставляют сделать выбор.

Далее эти данные передаются программе, которая их обрабатывает и выдает результат. Давайте задумаемся, как происходит процесс передачи.

Чтобы было легче понять его, я хочу начать с самого простого. Давайте попробуем передать данные скрипту без использования форм.

Хотя это будет очень не удобно, но сделать это возможно. И здесь нам поможет URL-адрес. Напомню, что URL-адрес это текстовая строка, которую вы пишете в адресную строку браузера.

Например:



Адрес: ← Это и есть URL-адрес → Переход

На самом деле URL может иметь немного другое представление, чем Вы привыкли его видеть. Он может быть более «длинным».

<http://www.bravoo.net/path/index.php?name=dima&age=21>

Как вы заметили, этот URL состоит из двух частей, которые разделены знаком «?».

Левая часть (<http://www.bravoo.net/path/index.php>) – путь к программе, которая будет все обрабатывать, а правая часть содержит параметры (переменные), которые передаются этой программе.

В данном случае файлу index.php будут переданы переменные: name=dima и age=21.

Если в этом файле написать код:

```
<?php  
Echo "Привет ".$name." тебе ".$age." год";  
?>
```

То результатом работы программы будет надпись «Привет dima тебе 21 год».

Но передавать данные таким образом очень не удобно, пользователю придется вручную набирать данные в адресную строку. Поэтому в веб-программировании очень часто используют формы. Они позволяют «спрашивать» данные у пользователя и передавать их программе в автоматическом режиме.

Обычно у формы, которая должна выполнять передачу данных скрипту есть выполняющая кнопка (в случае on-line голосования – это кнопка «Голосовать»). После того, как мы нажмем на нее, данные начинают передаваться одним из двух методов, которые мы можем указать в открывающем тэге <form> атрибутом method.

Есть два метода передачи данных GET и POST. Задать нужный нам метод мы можем примерно таким html-кодом:

```
<form method="POST">
```

Если надпись method="POST" НЕ указать, то по умолчанию всегда используется метод GET, а метод POST будет использован только тогда, когда его задать явно (method=POST).

Давайте разберемся, чем же отличаются два этих метода передачи данных. Рассмотрим это на простом примере.

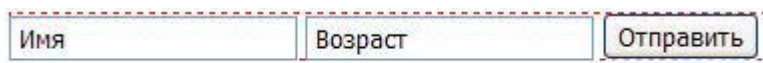
Допустим, у нас есть следующая форма, которая имеет два текстовых поля (для ввода имени пользователя и для ввода его возраста):

```
<form name="form1" method="get" action="www.bravoo.net/path/index.php">  
<input name="name" type="text" />  
<input name="age" type="text" />
```



```
<input type=submit name=go value='Отправить'>
```

```
</form>
```



Как видите в этой, в этой форме используется метод передачи данных GET. Если теперь нажать на кнопку «Отправить», то данные формы будут переданы в программу www.bravoo.net/path/index.php вместе с URL.

В нашем случае этот URL будет выглядеть следующим образом (в зависимости от тех значений, которые введет пользователь):

www.bravoo.net/path/index.php?name=имякотороевведетпользователь&age=возраст

Вся информация, которая передается таким способом, остается открытой и записывается в файл отчета на сервере.

В отличии от метода GET, метод POST помещает данные не в URL, а в тело запроса. Здесь мы можем передавать не только текстовую, но и любую другую информацию (графику, музыку, видео и др.). К тому же объем передаваемой информации может достигать 2 Гб, вместо длины максимального URL в методе GET.

Передаваемая информация таким способом нигде не записывается, соответственно этот метод более безопасен.

Так, что же лучше всего выбрать: POST или GET? В большинстве случаев применять можно, как тот, так и другой метод, все будет работать нормально. Однако, бывают такие ситуации, когда подходит только один из двух методов.

Рассмотрим некоторые из них, и я скажу, какой метод здесь предпочтительнее использовать:

- 1) Если ваша программа запускается с помощью ссылки, то лучше всего использовать метод GET.
- 2) В случае, когда Вы НЕ хотите, чтобы передаваемые вами данные записывались в файл отчета сервера, то используйте метод POST. Примером может быть ситуация, когда в форме вы указываете пароль. Очень не желательно, чтобы этот пароль хранился на сервере, правда?
- 3) Когда передавать нужно НЕ текстовую информацию используйте метод POST.

Глава 3. А знаете ли вы, что такое Cookies?

Ну, как слышали раньше такое определение? Оно очень часто используется в Интернет. Даже работая в обычном браузере мы можем видеть следующие опции «Включить или отключить cookies». Можете быть уверены, что если вы будете заниматься веб-программированием, то с этим понятием столкнуться придется еще не раз. Так, что же оно значит?

Мне бы хотелось начать с самого термина Cookies (на русском языке он произносится как «кукис»). Если перевести его с английского языка, то в буквальном смысле он переводится как «печенье». Почему это средство названо таким образом, я не знаю. Да и не в этом смысл, главное, что оно есть и работает, а остальное не так важно.

Вы когда-нибудь общались в Интернет на форумах? Если общались, то наверняка должны были заметить одну вещь. Когда **один раз** регистрируешься в нем и зайдешь в свой аккаунт, вас запоминают как пользователя. Теперь если вы закроете свой браузер и через некоторое время опять зайдете на этот форум, больше заходить в аккаунт (вводить свой логин и пароль) не надо, программа запомнила вас.

Как это достигается?

Я знаю два способа, как можно этого добиться.

1) Собирают IP-адреса каждого пользователя и помещают их в специальную базу данных. Если в будущем сайт посещает пользователь с определенным IP, программа сразу же заходит в его аккаунт.

Но этот способ оказался совершенно не работоспособным и на это есть две причины:

- **Причина 1.** Большинство пользователей не имеют фиксированного IP-адреса их компьютера. В первую очередь к таким людям относятся те, кто пользуется модемным Dial-up соединением. Каждый раз, когда такой человек подключается к Интернету, ему выдается новый IP-адрес, из числа не занятых у провайдера. Таким образом, под одним IP-адресом могут заходить сразу несколько человек.

- **Причина 2.** Если пользователей будет слишком много, то иметь такую базу данных и обрабатывать ее будет проблематично. Представляете, сколько места на жестком диске она будет занимать?

2) Второй способ основан совершенно на другом принципе и решает все проблемы. Здесь нам на помощь приходят Cookies! На самом деле Cookies представляет собой небольшую порцию информации, которая записывается в директорию браузера, которым вы пользуетесь. Заметьте, что эта информация хранится не на сервере форума, а на вашем компьютере. Таким образом, когда вы заходите на сайт вам присваивается определенная порция информации, которая записывается на вашем компьютере. В случае с форумом эта информация содержит в себе ваш логин и пароль. Теперь, когда вы заходите на сайт повторно, веб-программа считывает эти данные с вашего компьютера, и вы автоматически заходите как авторизованный пользователь.

Все гениальное просто! Эта информация и есть те самые cookies. Вы скажите: «Тогда я могу их сам удалить со своего диска?». Да, это на самом деле так. Решать, будут храниться Cookies на вашем компьютере или нет только вам. Именно поэтому в браузере есть опция включить и отключить Cookies.

Глава IV. Пишем программы на PHP сами

Глава 1. Формируем структуру сайта при помощи PHP.

У многих новичков сайтостроителей очень часто возникает одна неприятная проблема. Если они не находят ответ на эту проблему, то в дальнейшем она начинает забирать у них все больше и больше времени.

Итак, суть проблемы в следующем: у какого-то человека есть свой сайт, этот сайт состоит из 50 страниц. На каждой странице находится основное меню. А теперь представьте, что какой-нибудь пункт этого меню изменился. На всех 50-ти страницах придется каждый раз вручную вносить это изменение. Ситуация, честно сказать, не из приятных.

Неужели никак нельзя решить эту проблему? На самом деле существует один очень интересный выход из этой ситуации: нужно сделать меню сайта статичным, отдельным, неизменяемым файлом, а затем просто на него ссылаться из каждой из 50-ти страниц. Таким образом, меняя информацию только в одном файле с нашим меню, она сразу же изменится и на всех наших страницах.

Теперь самое главное решить, как это осуществить чисто с технической стороны. Я считаю, что самый приемлемый на сегодняшний день способ – это PHP. Сейчас почти все компании, которые предоставляют платный хостинг, поддерживают его. Кроме того, существует еще один способ – это использование фреймов. Но пользоваться данным способом я Вам не рекомендую. Сразу же начнутся проблемы с отображением страниц.

Итак, попробуем решить нашу проблему с помощью языка web-программирования PHP. Нам необходимо, чтобы меню было отдельным файлом. Его нужно будет вставить в соответствующее место в основном документе. Для этого создаем отдельный PHP файл и называем его menu.php. Создать файл с расширением *.php можно через html- редактор Dreamweaver. Заходим в меню «File/New../Dynamic Page/PHP».

Теперь нужно создать шаблон остальных страниц, которые будут использоваться на вашем сайте. Опять создаем PHP документ и назовем его, скажем page1.php. Туда вставляем основную структуру сайта, без меню. Примерно это может выглядеть вот так:

```
<html>

<head>

<meta http-equiv="Content-Language" content="en-us">

<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">

<title>Шаблон основной таблицы</title>

</head>

<body>

<table width=100% border=1 bordercolor=#000000 cellpadding="0">
```

```
<tr>

    <td>

        <table align=center>

            <tr>

                <td>

                    <img Ваша картинка>

                    <span lang="ru">Здесь располагается логотип</span></td>

            </tr>

        </table>

        <table border="0" width="100%" id="table1">

            <tr>

                <td width=25>

                    Сюда вставим наше меню.</td>

                <td><span lang="ru">Здесь будет располагаться
                    основной текст

                    страницы. Сюда можно помещать текст, рисунки,
                    ссылки или другие вложенные таблицы. При добавлении
                    нового материала, колонка автоматически становится
                    длиннее. </span></td>

            </tr>

        </table>

        <table border="1" width="100%" id="table2" cellspacing="0">

            <tr>

                <td>

                    <p align="center"><font size="2">Designed by Chengayev
                    Dmitriy, Copyright ©2006<br>

                    Частичная или полная перепечатка или иное использование материалов
                    сайта допускается только с разрешения автора. Ссылка на источник обязательна.
                    Нарушители преследуются по закону.

                    </font></td>

            </tr>

        </table>
```

```
</td>

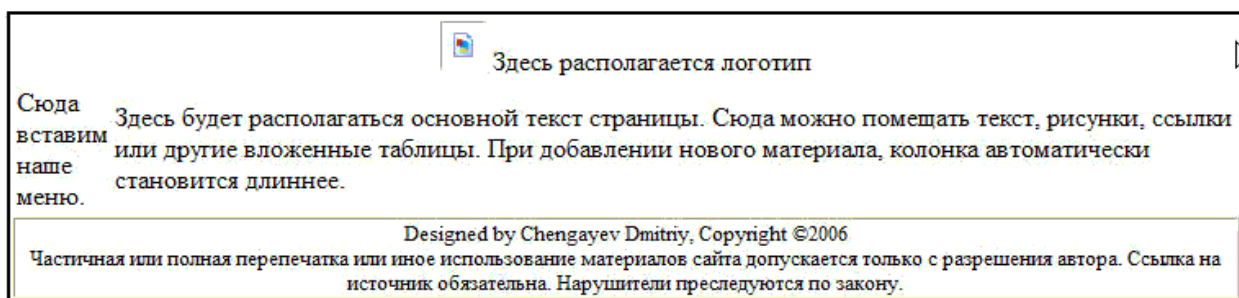
</tr>

</table>

</body>

</html>
```

У вас должна получиться страничка похожая на эту:



Ну и, наконец, самое главное: вставляем меню на соответствующее ей место. Делать это нужно при помощи PHP.

```
<tr>

<td width=25>

<?php //Сообщаем, что серверу, что мы будем использовать язык PHP.

Include "menu.php"

?>

</td>
```

Напомню, что **Include** – это одна из основных команд PHP. Ее синтаксис Include "имя файла.php". С помощью этой команды можно с легкостью вставлять содержимое файла, имя которого указывается, в то место, где располагается эта команда, при этом все коды PHP, содержащиеся во вставляемом файле, исполняются так же, как если бы они были на месте этой команды. Особенность здесь в том, что файл именно вставляется и пути к картинкам следует указывать от местонахождения того файла, в котором находилась команда include. Если вставляемого файла не существует, то программа выдаст соответствующее уведомление об этом.

Ну, вот и все, о чем я хотел вам рассказать. Применяя полученные знания на практике, теперь вам не нужно тратить свое драгоценное время на глупую. Желаю Вам успехов!

Глава 2. Как узнать сколько раз скачали ваш файл?

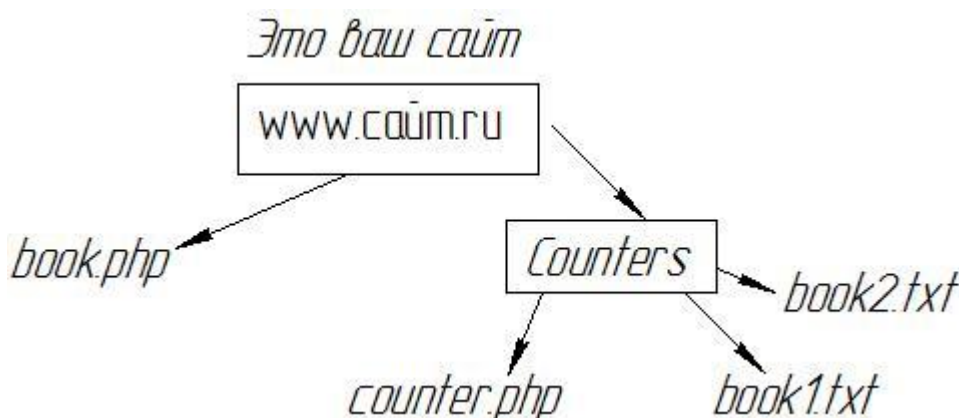
С того момента, как только будет создан ваш сайт, самым первым и очевидным желанием будет узнать, насколько он популярен. Существуют разные возможности, которые могут решить эту проблему. Обычно сервер, который предоставляет вам

хостинг, дает очень скудную статистику. В эту статистику входит только количество обращений на сайт и количество переданной информации. Но согласитесь, что этого совсем даже недостаточно. Файлов на сайте очень много и хотелось бы иметь представление о том, что же там у нас скачивают.

Ну и как тогда можно решить эту проблему? Выход есть - счетчики скачиваний. Не пугайтесь, все очень просто. Вы наверняка уже их видели. Обычно этот счетчик устанавливается рядом со ссылкой на скачиваемый документ. Примерно, все выглядит так: Ссылка – Скачали [656]. 656 – это и число, которое увеличивается на 1 при каждом скачивании файла, а программа, которая это осуществляет – счетчик скачиваний.

Очень важно понимать, как именно работает счетчик, а уже затем можно приступать к написанию кода.

Первым делом нужно определиться с его структурой: из каких файлов он будет состоять и как все будет взаимодействовать. Для наглядности я хочу привести следующий рисунок.



У нас есть какой-нибудь сайт `www.сайт.ru`. Допустим, что на этом сайте нужно подсчитывать сколько раз скачивают некоторые книги. Ссылки на книги находятся в файле `book.php`. Для удобства создаем отдельную директорию `Counters`, в которой будет сам счетчик (`counter.php`) и текстовые документы, в которых будут храниться данные, сколько раз скачали соответствующую книгу (`book1.txt` и `book2.txt`). Вы спросите, а почему бы не использовать для хранения числа скачиваний базу данных? Конечно, `book.txt` можно заменить базой данных, но для этого нужно специальное программное обеспечение. К тому же текстовые файлы использовать гораздо проще, чем базу данных. А я рассчитываю свою статью в первую очередь на новичков. Несомненно, что на большом сайте использование базы данных является необходимостью. Без нее там никак не обойтись. Просто количество текстовых файлов достигнет такого количества, что управляться с ними станет практически невозможным. Но это уже тема другой статьи.

В итоге, как результат, когда мы открываем файл `book.php`, то должны видеть примерно следующее:

Ссылка (Количество скачиваний: 25). 25 - это число, которое увеличивается на 1 при нажатии на ссылку.

Теперь попробуем описать, как это цикл работает в целом. Сначала загружается страница с нашей ссылкой и надписью – «Количество скачиваний: 25». Чтобы взять число 25 нужно обратиться к файлу, где оно храниться (book.txt). Как только пользователь нажимает на ссылку, запускается программа (counter.php), которая увеличивает переменную на 1 и записывает ее в текстовый файл. И так каждый раз.

Как видите теоретически все очень просто. Но основные проблемы возникают на практике.

Итак, давайте перейдем к технической стороне вопроса. Начнем непосредственно с самого счетчика (counter.php). Вот код, который необходимо в него вставить.

Counter.php

```
<?php

if ($book==1) {

    header("location: http://www.сайт.py/Zip/zip1.zip"); //Это файл, который нужно
скачать
$file=fopen("book1.txt","a+"); //Открытие
flock($file,LOCK_EX); //Блокировка
$count=fread($file,100); //Чтение
$count++; // Увеличение значение на 1
ftruncate($file,0); // Очищаем файл
fwrite($file,$count); //Записываем новое значение
flock($file,LOCK_UN); //Разблокируем
fclose($file); //Закрываем

}

else if ($book==2) {

    header("location: http://www.сайт.py/Zip/zip2.zip ");
$file=fopen("book2.txt","a+");
flock($file,LOCK_EX);
$count=fread($file,100);
$count++;
ftruncate($file,0);
fwrite($file,$count);
flock($file,LOCK_UN);
fclose($file);

}

?>
```

Ну, как, сложно? Давайте разбираться вместе.

Для начала определимся с конструкцией кода. Как вы знаете, чтобы использовать PHP, нужно создать следующую конструкцию:

```
<?php
```

```
...
```

```
?>
```

Тем самым мы даем программе понять, что используем PHP.

Так как скачиваются несколько файлов, чтобы определить на какую именно ссылку нажал пользователь, используется конструкция:

```
if ($book==1) {Счетчик 1} // Если $book==1, то выполняем Счетчик 1
```

```
else if ($book==2) {Счетчик 2} // Если $book==2, то выполняем Счетчик 2
```

```
else if ($book==3) {Счетчик 3}
```

и.т.д. Таким образом, отбор осуществляется с помощью переменной \$book. Единственное, каким образом сделать так, чтобы при нажатии на определенную ссылку, присваивалась соответствующая переменная?

Когда вы хотите перейти на определенный сайт, нужно набрать его URL.

Например:

```
www.сайт.ru/index.php
```

Но на самом деле URL может иметь более длинный вид. Видели ли вы такую запись:

```
www.сайт.ru/index.php?параметры
```

Здесь URL, как бы, разбивается на 2 части: в первой, находится адрес к сайту, а во второй (после знака «?») указываются некоторые параметры. Разделителем является знак вопроса.

Этими параметрами могут быть переменные.

```
www.сайт.ru/index.php?переменная=значение
```

Указывая, так переменные, они будут передаваться в открываемый файл. В случае, если нужно объявить сразу несколько переменных, делается это через знак «&».

```
www.сайт.ru/index.php?переменная1=значение&переменная2=значение
```

Таким образом, когда пользователь открывает данную страницу, переменные присваиваются автоматически, что очень удобно. Такой метод передачи переменных в сценарий называется методом GET.

Применим данный метод и в нашем случае. Нужно зайти в документ со ссылками (book.php) и изменить ссылку на файл, следующим образом:

```
<a href="Counters/counter.php?book=1">. Т.е. сначала идет ссылка на файл счетчик, а после знака «?» задается имя переменной book, которая затем и
```


направит пользователя к соответствующему файлу. Аналогично поступаем для второй и третьей ссылки.

```
<a href="Counters/counter.php?book=2">
```

```
<a href="Counters/counter.php?book=3">
```

Вот и вся конструкция теперь можно переходить к коду самого счетчика.

Разберем код-счетчика.

- `header("location: http://www.сайт.py/Zip/zip1.zip");` - собственно это и есть ссылка на книгу, которую необходимо загрузить. `header()` – это специальная функция, которая может перейти на новый web-узел, не спрашивая об этом пользователя. Но использовать данную функцию нужно осторожно. Основная ее особенность в том, что она не может запуститься после HTML кода. Иными словами, нельзя сначала поместить html-код, а затем, где-то в середине, `header()`.

```
- $file=fopen("book.txt","a+");
```

Как уже говорилось выше, для того, чтобы хранить переменную количества скачиваний, используется текстовый файл. Собственно вся задача счетчика заключается в том, чтобы открыть этот файл, узнать, какое значение сейчас у счетчика, увеличить это значение на 1 и закрыть файл. Вот и все!

В PHP, для работы с текстовыми файлами, обязательно выполнять следующие этапы:

- 1) Открыть файл
- 2) Извлечь данные из файла и при необходимости их изменить
- 3) Закрыть файл

Специальная функция для открытия файла: `$переменная = fopen("имя файла", "режим");`

При открытии файла нужно указать режим, который будет разрешать выполнять с файлом определенные действия после его открытия. Режимы, которые можно устанавливать приведены в следующей таблице:

Режим	Что будет, если файл существует	Что будет, если файла не существует
r	Только чтение	Возникнет предупреждающее сообщение
r+	Чтение и запись	Возникнет предупреждающее сообщение
w	Только запись	Будет создан новый файл. Если файл существует, новый файл будет записан поверх

		старого
w+	Чтение и запись	Будет создан новый файл. Если файл существует, новый файл будет записан поверх старого
a	Добавление данных в конец файла	Будет создан новый файл.
a+	Чтение и добавление данных в конец файла	Будет создан новый файл.

Таким образом, мы открываем файл book.txt для чтения и изменения его значения. Создается специальная переменная \$file, в которой будет храниться открытый файл.

- flock(\$file,LOCK_EX);. Очень важный элемент. Он блокирует доступ к файлу, на время работы скрипта. Представьте себе, что на ссылку одновременно нажали сразу же два человека. Так как скрипт открывает файл для записи, то может произойти сбой и счетчик просто обнулиться.

- \$count=fread(\$file,100); Далее считываем информацию с открытого файла \$file и присваиваем ей переменную \$count.

- \$count++; - увеличиваем значение на 1. Эта запись эквивалентна следующей (\$count = \$count+1).

- ftruncate(\$file,0); - с помощью этой команды содержимое файла (book.txt) очищается. И теперь можно будет свободно записывать туда новую информацию. Именно на этом моменте может произойти сбой при запуске одновременно двух скриптов. Об этом я говорил выше. Но нам уже ничего не страшно, файл заблокирован.

- fwrite(\$file,\$count);. Записываем новое значение счетчика.

- flock(\$file,LOCK_UN);. Теперь файл можно разблокировать и

- fclose(\$file);. Закреть.

Когда все написано, не забудьте установить атрибуты (права) на текстовые файлы (txt). Сделать это можно через FTP-клиент. Изменяем атрибуты файла на 666 (rw-rw-rw).

Ну, вот и все. Я старался как можно более понятно и просто объяснить вам процесс установки счетчика скачивания. Тем не менее, если у вас, что-нибудь не получается, пишите мне, и мы постараемся решить проблему вместе.

Глава 3. Счетчик посещений.

Если вы регистрировались на каком-нибудь рейтинге, то наверняка уже знаете, что такое счетчик посещений. Проще говоря, это некая картинка, которая

располагается в определенном месте вашего сайта и показывает цифру - число посещений страницы. Счетчики посещений могут быть оформлены по-разному, но суть везде одна.

В данной статье я не буду рассматривать навороченных счетчиков посещений, а расскажу лишь о принципе их работы. Прочитав эту статью, вы сможете сразу же приступить к установке такого счетчика на вашем сайте.

Данный счетчик написан на PHP. Его задача считать сколько раз была открыта данная страница, и выводить результат в виде «Данную страницу прочитали N раз».

Такой счетчик очень удобно использовать для того, чтобы узнать сколько раз была прочитана ваша статья. Для удобства внизу страницы можно расположить надпись «Данную статью прочитали N раз». Еще раз напоминаю, что это всего лишь образец и никакого оформления здесь нет. На основе этого кода вы сможете усовершенствовать его так, как вам понравится.

Принцип действия этого счетчика очень прост. У нас есть какое-то начальное число, которое показывает, сколько раз была открыта данная страница. При каждом новом открытии страницы это начальное число увеличивается на 1.

Согласитесь, что начальное число должно где-то храниться. Для этой цели будем использовать обычный текстовый файл (data.txt). Создайте его в той же директории, где находится сама статья.

Далее все просто: на странице с нашей статьей размещаем следующий код:

```
<?
$file=fopen("data.txt","a+");
flock($file,LOCK_EX);
$count=fread($file,100);
$count++;
ftruncate($file,0);
fwrite($file,$count);
flock($file,LOCK_UN);
fclose($file);
echo "Эта статья была прочитана $count раз";
?>
```

Задача счетчика посещений показать сколько раз была просмотрена данная страница. Я не буду приводить пояснений, как работает этот код. Здесь используются такие же функции, как и в счетчике скачивания. Если, что-то непонятно, обратитесь к предыдущему примеру.

Глава 4. Проводим свое on-line голосование.

Наверняка, когда-нибудь вы хотели узнать мнение посетителей по какому-нибудь вопросу. Голосование – это очень мощный инструмент для выполнения данной задачи. Начнем с того, зачем вообще нужно голосование? Для начала, можно определить портрет своей аудитории (возраст, профессию, уровень заработка, и.т.д). Узнав этот портрет, нужно соответствующим образом менять дизайн, стиль и контент вашего сайта. Кроме того, ваш сайт становится интерактивным. Доказано, что интерактивные сайты пользуются большей популярностью у посетителей. Да, в конце концов, вы сами можете придумать, как можно использовать этот мощный инструмент.

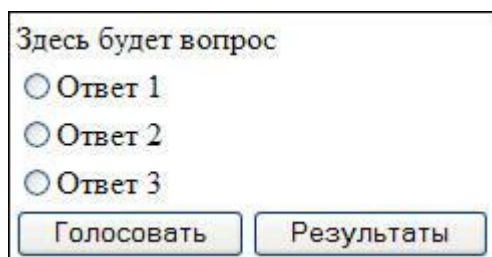
Итак, нам нужно установить систему голосования на сайт. Где ее взять? Самый легкий вариант для установки такой системы, это воспользоваться бесплатными системами голосования. Такие системы можно найти на специализированных сайтах. Вам предлагают установить соответствующую форму на сайт. Но это просто форма, вся ее обработка уже будет производиться не на вашем сайте, а на удаленном сервере и это вносит некоторые недостатки в работу голосования. Во-первых, настройка такого голосования происходит On-line и занимает достаточно много времени. Во-вторых, вы не можете контролировать работоспособность этой системы, все зависит от удаленного компьютера.

Именно поэтому, я предлагаю написать свой собственный скрипт голосования на PHP. Конечно, этот вариант более сложный, но зато хозяин теперь вы и никто больше не будет диктовать вам свои условия.

Давайте определяться, что нам может понадобится для работы голосования. Требований не так уж много: поддержка PHP и 3 файла. Почему именно три? В первом будет располагаться сама форма с вопросом. Назовем его (Index.php). Во втором будем хранить результаты голосования (opros.txt). А третий – это сам скрипт, который обрабатывает данные и выводит результаты (opros.php). Также, когда отправите эти файлы на сервер, не забудьте установить соответствующие права для файла opros.txt. Установите для него атрибут 666. После этого, он станет доступен для записи. Напоминаю, что атрибуты можно установить с помощью FTP – клиента.

Index.php

Для начала займемся формой, где будет сам вопрос, варианты ответов, и кнопки (Отправить и Результаты). В принципе, данную форму можно вставлять в любой файл вашего сайта. Примерно форма может выглядеть так:



Здесь будет вопрос

☐ Ответ 1

☐ Ответ 2

☐ Ответ 3

Это только набросок. Все, что касается дизайна и оформления, прошу строго не судить. В задачи данной статьи я ставлю лишь обеспечить работоспособность опроса. Так, что вопросы оформления оставляю для вашей личной доработки. Количество ответов вы сможете также настроить сами. Так, что не пугайтесь, что пока здесь всего три ответа. Вот примерный код, для создания такой формы.

```
<FORM METHOD="POST" action="opros.php">  
  
Здесь будет вопрос  
  
<INPUT TYPE=radio NAME=otvet VALUE=1>Ответ 1<br>  
  
<INPUT TYPE=radio NAME=otvet VALUE=2>Ответ 2<br>  
  
<INPUT TYPE=radio NAME=otvet VALUE=3>Ответ 3<br>  
  
<INPUT TYPE=Submit VALUE="Голосовать">  
  
<INPUT TYPE=Submit VALUE="Результаты">  
  
</FORM>
```

Методом POST данные формы передаются скрипту opros.php. В ней имеются 2 кнопки: голосовать – для голосования, результаты – для просмотра результатов без голосования. В дальнейшем вы можете изменить оформление, так как вам понравится. Я привожу лишь набросок.

Opros.txt

Теперь нам нужен файл, где будет храниться полученная от пользователя информация. Легче всего использовать для этой цели обычный txt файл. Создайте его и введите туда следующие данные.

```
Ваш вопрос? (Строка 0)  
  
0 (Строка 1)  
  
0 (Строка 2)  
  
0 (Строка 3)
```

Очень важно расположить все по соответствующим строкам. Номер строки (например, «Строка 0») писать не обязательно. Это я для наглядности.

Opros.php

Ну и напоследок самое сложное и интересное - скрипт, который все обрабатывает.

```
<?php  
  
$file = "opros.txt";  
  
$masiv = file($file);
```

```
$i = 1;

$n = 0;

setcookie("golos","yes");

$kolelementov = Count($masiv);

while ($i <= $kolelementov):

$masiv[$i] = trim(str_replace ("\n","", $masiv[$i]));

$n = $n+$masiv[$i];

$i++;

endwhile;

if ($otvet!= "" && ($golos != "yes"))

{

echo "<br><b>Спасибо, Ваш голос принят</b>";

$masiv[$otvet]++; //Увеличиваем значение соответствующего элемента на 1.

$n++; // Количество проголосовавших тоже увеличивается на 1.

$result = "Какая тема больше всего интересна для вас?\n".$masiv[1]."\n".$masiv[2]."\n".$masiv[3]."\n"; //Переменная с результатом.

$foopen = fopen($file,"w");//Открываем файл для записи.

if ($foopen) // Если файл открылся

{

fwrite($foopen,$result); //Записываем в него результат

fclose($foopen); //Обязательно закрываем

}

else { echo "Ошибка при открытии файла"; }

}

//Иначе, если значение cookies уже есть, то не разрешаем больше голосовать и выводим результаты.

else {

if ($golos == "yes") {

echo "<b>Спасибо! Ваше мнение уже учтено. Вы не можете больше голосовать!</b>";
```

```
}  
  
echo "<br>Результаты голосования";}  
  
echo "<br>Основы. HTML. — <b>".$masiv[1]."</b>";  
  
echo "<br>CSS. Верстка. Стили. — <b>".$masiv[2]."</b>";  
  
echo "<br>Web-программирование. PHP. — <b>".$masiv[3]."</b>";  
  
echo "<br><br>Всего голосов: ".$n;  
  
echo "<br><br>";  
  
?>
```

<p>На главную</p>//Ну и ссылочка, чтобы пользователь мог //вернуться назад.

Давайте разбираться, что это мы здесь написали.

\$file = "opros.txt"; - переменной \$file присваиваем значение opros.txt (файл, где хранятся результаты голосования). Теперь, один раз изменив эту переменную, больше ничего нигде менять не надо.

```
$masiv = file($file);
```

Чтобы осуществить процесс голосования, будем использовать массив. Но для начала этот массив нужно заполнить соответствующими значениями. Если вы когда-нибудь занимались программированием, то должны знать, что массив можно заполнить с помощью цикла. Примерно это может выглядеть следующим образом:

```
$forn = fopen("file.txt", "r");//Открываем текстовый файл с массивом для чтения с него.
```

```
While(!feof($forn)) // Пока не будет достигнут конец файла, делать
```

```
{
```

```
    $masiv[] = fgets($forn);//Присваивать каждому элементу массива значение  
    //соответствующей строки
```

```
}
```

```
Fclose($forn);//Закрываем открытый файл.
```

Результатом всего, что мы здесь с вами написали, будет одномерный массив \$masiv. Количество элементов этого массива будет равно количеству строк в текстовом файле, а значение каждого элемента содержит значение соответствующей строки.

Чтобы каждый раз не писать эту сложную конструкцию, разработчики PHP ввели специальную функцию file(), которая осуществляет те же самые действия. В результате работы этой функции имеем тот же самый массив.

```
$masiv = file("file.txt");
```

Согласитесь, что мнение одного человека нам нужно только один раз. Зачем нам, чтобы один и тот же человек голосовал 1000 раз. Это недостаток мы сейчас и попытаемся устранить. Проблема в том, как определить, что голосовал один и тот же человек. Можно, конечно, вычислять посетителя по его IP, но этот способ не совсем совершенен. Дело в том, что те люди, которые пользуются dial-up, не имеют постоянного IP. Стоит им только снова перезвонить провайдеру и их IP уже будет другой. Но к счастью есть более продвинутое средство – cookies. Оно представляет собой пару «переменная=значению». Но главная особенность cookies – они сохраняются браузером на компьютере пользователя. Таким образом, когда пользователь, следующий раз зайдет на страницу, то его уже можно будет свободно вычислить.

```
setcookie("golos","yes");
```

setcookie() специальная функция для установки cookies. Мы воспользуемся следующими ее параметрами:

```
setcookie(«переменная», «значение»);
```

Таким образом, если следующий раз пользователь зайдет на вашу страничку и запустит скрипт, но у него будут записаны cookies, то проголосовать он не сможет.

Далее, вводим необходимые для скрипта переменные.

`$i = 1;` Нельзя забывать, что отсчет в PHP ведется с нуля. Чтобы не путаться с этим, мы не будем использовать нулевую строку и считать будем с 1. В этом нам поможет переменная `$i`.

`$n = 0;` Это переменная – количество проголосовавших людей. Ее необходимо обнулять при каждом запуске цикла. Почему? Чуть позже поймете.

```
$kolelementov = Count($masiv);
```

Count() – это функция, которая определяет, сколько элементов в массиве и возвращает результат в виде числа.

```
while ($i <= $kolelementov):
```

```
$masiv[$i] = trim(str_replace ("\\n","", $masiv[$i]));
```

```
$n = $n+$masiv[$i];
```

```
$i++;
```

```
endwhile;
```

Пока номер проверяемого элемента (начиная с 1), не последний (включительно), делаем следующие действия:

```
$a[$i] = trim(str_replace ("\\n","", $a[$i]));
```

Чтобы обрабатывать текстовую информацию, в язык PHP встроено множество специальных функций. Одну из таких функций мы используем в нашей системе

голосования. При работе скрипта, в текстовом файле, где хранятся данные голосования, могут появиться нежелательные пробелы. Эти пробелы могут нарушить нормальную работу программы. Поэтому их нужно удалить. Для этой цели существует функция trim().

`$string = trim($string)`. Здесь, у строки `$string` будут удалены лишние пробелы в начале и в конце строки.

Еще одна функция работы с текстом, которую нам придется использовать `str_replace()`. Принцип ее действия достаточно прост:

`str_replace("a","b","строка")`; В данной «строке» все фрагменты «a» заменяются фрагментами «b».

И, наконец, символ «`\n`» - это принудительный переход на новую строку.

`$n = $n+$a[$i]`;

Количество скачиваний – это сумма всех элементов массива. Именно поэтому `$n` нужно каждый раз обнулять, иначе все будет прибавляться к старому значению.

`if ($otvet!= "" && ($golos != "yes"))`

«`!=`» - это знак неравенства. Если переменная «ответ» не пустая, т.е. кто-то ответил,

«`&&`» - И, переменная «голос» (из cookies) не «yes», то выполняем следующие действия.

Далее в скрипте используются достаточно простые и однотипные команды. Что они делают, вы можете посмотреть в соответствующих комментариях расположенных рядом с ними.

ЗАКЛЮЧЕНИЕ

Ну, вот, и закончились страницы этой электронной книги. Я не знаю, насколько она оказалась для вас полезной. Искренне надеюсь, что после ее прочтения у вас на сайте уже появился (а может быть не один) PHP-скрипт.

Как видите, PHP дает огромные возможности для развития сайта. Область его применения ограничивается только вашей фантазией. Можете считать данную книгу первой ступенькой в огромный мир веб-программирования. Если вы сумели вступить на нее, то сможете вступить и на остальные.

Если в ходе прочтения материала у Вас появились какие-нибудь вопросы или проблемы, обязательно пишите мне info@bravoo.net. Буду рад оказать помощь.

Кстати, у меня есть к вам одна небольшая просьба: если вам не трудно, напишите мне, что вам понравилось, а что нет в этой книге. Я учту ваши пожелания и постараюсь исправить недостатки.

Еще раз напоминаю, если Вам понравилась данная книга, то Вы можете совершенно бесплатно разместить ее на своем сайте. Подумайте, если она оказалась полезной Вам, вполне возможно она окажется полезной и посетителям Вашего сайта. А если посетители находят на сайте, что-то полезное у них появляется свойство возвращаться на него. Завлекайте их на свой сайт проверенными способами.

Благодарю, вас за проявленное внимание и на этом прощаюсь с Вами. Мне остается только пожелать Вам успехов.

Всего доброго! Всегда жду Вас на моем сайте.

С Вами был Ченгаев Дмитрий.

<http://www.bravoo.net> – Техническая информация новичку для создания сайта.

До свидания!

Электронный курс: "Практика Создания Реального Сайта".



"Узнайте Секреты Профессиональных Веб-Мастеров И Уже Через 1 Месяц Вы Будете Иметь Мощное Средство Для Общения, Заработка и Личного Развития - Собственный Веб-Сайт"

"Пошаговые инструкции в виде тренинга, более 6 Мб текстовой и графической информации (более 100 иллюстраций) + 20 Мб Видео-Уроков помогут осуществить мечту в реальность."

А знаете ли вы...

- Какой хостинг (место, где будет храниться сайт) лучше всего подойдет именно Вам?
- Нужно ли знать HTML или можно обойтись конструктором сайтов или CMS? Что такое лог-файлы?
- Где взять и как установить CMS?
- Какое программное обеспечение Вам понадобится и где его можно взять?
- Команды HTML и CSS и умеете ли правильно применять их на практике?
- Как правильно расположить все файлы на сервере, чтобы потом было намного легче?
- И многое другое

Ответы на эти вопросы вы можете найти на этой странице

<http://www.bravoo.net/tovar/index.php>