

К ВОПРОСУ СОЗДАНИЯ ДИНАМИЧЕСКИХ WEB-СТРАНИЦ

Б.А. Матюшкин¹⁾, К.В. Часов²⁾

1) студент Армавирского механико–технологического института (филиала) ФГБОУ ВО «Кубанский государственный технологический университет», г. Армавир, Россия, nick.bogdan2012@mail.ru .

2) к.п.н., доцент кафедры общенаучных дисциплин Армавирского механико-технологического института (филиала) ФГБОУ ВО «Кубанский государственный технологический университет», г. Армавир, Россия, chasov_kv@mail.ru .

Ключевые слова: интернет, сайт, динамический сайт, web-сайт, web-страница, создание web-страниц, язык PHP, скрипт, index.php.

Аннотация: в статье исследуются методы создания динамических web-страниц с помощью языка программирования PHP и написание скриптов на этом языке.

TO THE QUESTION OF CREATING DYNAMIC WEB PAGES

B.A. Matyushkin¹⁾, K.V. Chasov²⁾

1) the student Armavir mechanics-technological Institute (branch) Kuban state technological University, city of Armavir, Russia, nick.bogdan2012@mail.ru .

2) Ph. D., associate Professor, Armavir mechanics-technological Institute (branch) Kuban state technological University, city of Armavir, Russia, chasov_kv@mail.ru.

Keywords: internet, site, dynamic site, web-site, web-page, creating web-pages, PHP language, script, index.php.

Abstract: the article investigating methods of creating dynamic web-pages with programming language PHP and creating scripts on this language.

Последние десятилетия продемонстрировали социуму просто невероятное развитие Интернета и новых способов общения между людьми. Наиболее важной частью этого развития явилась World Wide Web (WWW) – Всемирная паутина. Каждый день в интернет появляются новые сайты, а пользователям предлагается и даже навязывается всё большее количество самых различных видов услуг, в том числе новых. Такое развитие информационно-технологического рынка неизбежно ставит перед

специалистами задачу улучшения и развития системы маркетинга, связанного с использованием интернет-технологий. В результате новые web-технологии претерпевают интенсивное развитие, а разработчики этих технологий получают значительный пакет заказов. Язык программирования PHP является одним из ярчайших представителей указанных технологий.

PHP (Hypertext Preprocessor) - это язык, предназначенный для быстрого создания динамических web-страниц. Отличие динамических сайтов от статических заключается в возможности интерактивного изменения внешнего вида страницы, что делает динамические сайты намного функциональнее статических. Динамические сайты во всем превосходят статические – высокая скорость добавления нового контента разработчиков, наличие «диалогов» с пользователем и отсутствие необходимости владения веб-мастерингом (HTML, CSS). Это обусловило востребованность разработчиков, владеющих языком PHP.

Синтаксис языка похож на синтаксис таких языков программирования, как C, Java и Perl, имея при этом свои собственные отличия. В официальной документации PHP определяется как server-side HTML-embedded scripting language (встраиваемый в HTML скриптовый язык с обработкой на сервере). Из этого определения можно выделить два основных момента:

– обработка кода PHP производится на сервере, а не в браузере клиента (как, например, в JavaScript);

– код PHP может быть встроен непосредственно в тело HTML страницы, что выгодно отличает его от Perl.

Основное преимущество PHP – простота, гибкость и скорость выполнения.

Ниже приведен пример PHP скрипта, взятый из фирменной документации по PHP, код PHP выделен курсивом.

```
<html>  
<head>  
<title>Example</title>  
</head>  
<body>  
<?php echo "Hi, I'm a PHP script!"; ?>  
</body>  
</html>
```

Этот пример иллюстрирует принцип встраивания кода PHP в HTML код страницы: все, что находится между начальным и конечным тегом PHP (<?php и ?> соответственно) считается PHP кодом и обрабатывается интерпретатором PHP.

Среди методов программирования сайтов средствами PHP отметим следующие методы: объектно-ориентированный, модульный и др.

Объектно-ориентированное программирование (ООП) – один из самых важных составляющих частей PHP, т.к. позволяет программисту моделировать в создаваемых приложениях реальные объекты и ситуации. Отметим при этом, что PHP не является полноценным объектно-ориентированным языком (например, таким, как C++ или Java), оставаясь вполне самостоятельным в смысле наиболее общих возможностей объектно-ориентированного программирования.

Язык PHP постоянно развивается, и практически в каждом улучшении мы можем заметить всё большую оптимизацию работы с классами и объектами, да и вообще с возможностями ООП. 90% улучшений связано именно с ООП.

Модульный метод получил большее распространение. Приглядимся к этому методу поближе. Общая схема такого подхода такова: существует некий главный скрипт, например `index.php`. Этот скрипт в некоем параметре, например, `$mod`, принимает имя модуля, который отдаёт браузеру какую-нибудь страницу.

Для начала необходимо создать систему каталогов сайта. Например такую, которая чаще всего используется программистами:

`/mod/` - каталог для хранения модулей;

`/img/` - картинки;

`/inc/` - каталог вспомогательных файлов;

В корневом каталоге хранится всего один файл – `index.php`. Это и будет самый главный представитель сайта.

Составляется дизайн сайта – общая процедура разработки любого сайта. В качестве примера: верх, низ, главное меню слева, а остальная часть – содержимое подключаемого модуля. Для этого в каталоге `inc` создается три модуля, названия которых вполне говорят сами за себя и отвечают за соответствующую часть сайта на экране: `top.php`, `bottom.php` и `leftmenu.php`, описывающие свои части.

В этом случае подключаемый модуль будет выглядеть следующим образом.

```
<?php
$page_title="Демо-сайт про php";
include ("inc/top.php");
include ("inc/leftmenu.php");
// информационная часть модуля
include ("inc/bottom.php");
?>
```

Приведём содержание главного скрипта.

```
<?php
include ("inc/config.php"); // подключение файла конфигурации
include ("inc/functions.php"); // подключение функций, используемых в модулях
// проверка на существование запрашиваемого модуля
if (!isset($mod) || ($mod=="") || (!file_exists("mods/$mod.php")))
{
    // если запрашиваемого модуля нет, то включить модуль по умолчанию
    $mod=$sys_def_mod;
}
include ("mods/$mod.php"); // подключение модуля
?>
```

Файл конфигурации может содержать наиболее используемые системные переменные, такие как имя модуля по умолчанию, адрес сайта, адрес электронной почты web-мастера, а также команды на подключение базы данных и т.д.

Для вызова модулей применяется команда, записанная с помощью строки `index.php?mod=имя_модуля`.

Приведём простейшую схему подготовки и использования модулей. Любой модуль добавляется в систему просто перемещением его в каталог `mods/` и пользуясь общими правилами работы с РНР возможно построение довольно сложного и разветвлённого сайта. Рассмотрим преимущества подобной работы.

- Разработчик практически не занимается привязкой (применением) кода к дизайну, выполнив указанное один раз в `index.php`. При этом модуль просто работает на благо сайта и разработчика.

- Централизация сбора основной информации из базы или конфигурационного файла, глобальных переменных сайта, информации о пользователе и т.д.

Необходимо отметить, что, как и в любой системе программирования (разработки), в РНР имеются недостатки, в частности, приходится чётко отслеживать какие имена переменных используются до модуля, чтобы не перезаписать, случайно, их внутри модуля. Правда, это настолько несущественный недостаток, что не имеет смысла обращать на него внимание.

Для информационной образовательной сред кафедры приведённая среда разработки подходит больше всего, т.к. обладает широчайшими возможностями для расширения сайта, дополнений и добавлений Web-страниц с необходимой кафедре тематической информацией. Обучающиеся – студенты-программисты – могут на практике помочь кафедрам института в создании не только сайта кафедры, но и всех необходимых сервисов, обеспечивающих функционирование кафедральной базы знаний,

документации и т.п. Для студентов указанная деятельность будет творческой, развивающей их способности.

Список использованных источников:

1. Горovenko Л.А. Экспертная оценка электронного программно-методического комплекса // Научные труды Кубанского государственного технологического университета. - 2014. № 54. С.355-361.

2. Вотякова В.С., Часов К.В. Включение обучающих интерактивных документов по математике в информационную образовательную среду // Успехи современного естествознания. – 2013. – № 10. – С. 104-105; URL: <http://www.natural-sciences.ru/ru/article/view?id=32986> (дата обращения: 3.10.2018).

3. Смольняков И.М., Часов К.В. Некоторые свойства прогрессирующих последовательностей // Международный журнал экспериментального образования. – 2014. – № 7-1. – С. 106-107; URL: <http://www.expeducation.ru/ru/article/view?id=5515> (дата обращения: 14.08.2018).

4. Паврозин А.В. Формирование профессиональных компетенций на примере подготовки бакалавров в техническом вузе // Электронный сетевой политематический журнал «Научные труды КубГТУ», Выпуск № 4, 2014.

5. Паврозин А. В., Филимонов В. В. Возможности языка C# в создании тестов // Международный студенческий научный вестник. 2016. № 5-3. С. 361-364; . URL: <http://www.eduherald.ru/ru/article/view?id=15948> (дата обращения: 02.11.2016).