4. СРЕДСТВА СОЗДАНИЯ WEB-СТРАНИЦ

[скрипт, PHP, скриптовый язык, сервлет, фреймворк]

4.4. Web-программирование на стороне Web-сервера

В этой части темы рассмотрены способы генерации Web-документов. Как уже отмечалось ранее, Web-документы могут быть представлены в двух видах:

- уже размеченной Web-страницей;
- Web-страницей, генерируемой сервером с помощью:
  - вызова CGI-совместимого шлюза-программы;
  - отдельного программного модуля.

Размеченный Web-документ может быть составлен по схеме, описанной далее в разделе 4.5, и просто положен в папку (директорию на Web-сервере), из которой Web-сервер раздает файлы. Для создания генерируемых файлов пишется специальный *скрипт*[1] использующий CGI, *FastCGI*, *WSGI* или иной интерфейс предоставляемый модулем Web-сервера, или же загружаемый из контейнера, соединенного с Web-сервером.

4.4.1. Серверные скрипты

Наиболее известными серверными скриптами являются счетчики посещений, гостевые книги, системы голосований и т.п.

Серверные скрипты представляют собой программы, чаще всего, написанные на языке *PHP*, выполняющиеся на стороне Web-сервера. Основное отличие серверных скриптов от пользовательских – возможность записи в файл, размещенный на Web-сервере. PHP – серверный скриптовый язык, инструкции которого можно внедрять в HTML-документы. Эти инструкции (программный код), выполняясь на стороне Web-сервера, выдают результат обработки данных непосредственно в окно Web-браузера. В виду этой особенности язык PHP также называют препроцессором HTML. Язык PHP представляет из себя достаточно гибкий язык, оснащенный возможностями работы с MySQL, протоколами *NNTP*, *POP3*, *IMAP* и многими другими. При помощи PHP можно реализовать весь функционал, который обеспечивают CGI-приложения, отличие состоит лишь в том, что PHP значительно проще для понимания и самостоятельного изучения.

Другим широко распространенным вариантом серверных скриптов являются CGI-скрипты. Как правило, CGI-скрипты пишутся на языке *Perl* (*Перл*), поэтому их иногда называют *Perl-скриптами*.

4.4.2. Использование CGI

CGI шлюзом может быть любая программа, которую вызывает Web-сервер. Для его работы достаточно напечатать на вывод, в самом начале, строку:

```
Content-type: text/html\n\n ,
```

где **\n** - символ новой строки. Далее можно выводить код страницы, используя при этом

---

1 *Скрипт – программа, выполняемая на Web-странице по запросу посетителя Web-сайта. Обычно язык скриптов включает простые структуры управления: линейные последовательности, циклы и условные выражения.*
В сетевых технологиях различают скрипты клиентской и серверной сторон.

возможности выбранного языка, например, используя язык программирования *Python*:

```
print "Content-type: text/html"
print
print """
<html>
  <head>
    <title>Web-страница с CGI</title>
  </head>
<body>
"""
for i in range(5):
  print "<p>Переменная i = %s</p>" % i
print """
  </body>
</html>"""
```

Content-type – это, так называемый, «*заголовок*». Существуют и другие, часто используемые заголовки: *Expires*, *Last-modified*, *Set-Cookie*.

При вызове шлюза Web-сервер устанавливает некоторые переменные среды (*environment variables*), которые можно получать стандартными средствами языка, например следующим образом можно получить строку запроса на Python:

```
import os
print "Content-type: text/html"
print
print """
<html>
  <head>
    <title>Web-страница с CGI</title>
  </head>
  <body>
    <p>%s</p>
  </body>
</html>""" % os.environ["QUERY_STRING"]
```

Для большинства скриптовых языков программирования (Perl, PHP, Python) существуют модули работы с CGI.

CGI – базовый принцип взаимодействия Web-сервера и разрабатываемого приложения.

4.4.3. Использование FastCGI

Поскольку при каждом запросе программа-шлюз запускается заново, это часто приводит к снижению производительность. Поэтому часто используется *FastCGI* – развитие CGI, использующее уже запущенные процессы несколько раз. Для этого применяется несколько иной подход – Web-сервер единожды запускает программу-шлюз, которая работает постоянно. При получении запроса он соединяется с программой с помощью *сокета*[2] или по протокольному стеку TCP/IP, передает ей данные. В ответ эта программа возвращает результат – сформированную Web-страницу. Написание своего сокета сервера – довольно сложная задача, поэтому часто используются библиотеки, поставляемые со скриптовым языком.

4.4.4. Использование отдельных программных модулей

Программные модули являются весьма эффективными, так как при своем старте они загружают в память интерпретатор языка и используют его для обработки скриптов.

---

2 *Сокет (в рассматриваемом случае) – программный интерфейс, обеспечивающий обмен данными между процессами (приложениями).*

Для каждого скриптового языка программирования они различны:

- Python – обычно используется модуль *mod_wsgi*, реализующий *WSGI* (*Web Server Gateway Interface*); позволяет «встроить» интерпретатор или использовать сетевой режим, как FastCGI. Интересная особенность – наличие так называемого *middleware*-компонента в цепочке шлюз-сервер, который является для шлюза – сервером, а для сервера – шлюзом. Это позволяет создавать длинные цепочки python скриптов для генерации Web-страницы.

- PHP – чаще всего используется *mod_php*. Но из-за большого объема памяти, занимаемого им, иногда применяют FastCGI. Существует также отдельный модуль *php-fpm* (*PHP FastCGI Process Manager*) – улучшение FastCGI.

- Perl – используется *mod_perl*. Существует аналог WSGI под названием *PSGI*.

- Ruby – решение, аналогичное WSGI, – *Rack*. Модуль для сервера – *Phusion Passenger*; поддерживается также и языком Python с помощью WSGI и *Node.js*.

4.4.5. Контейнер сервлетов

Контейнер сервлетов[3] – программное обеспечение, представляющее Web-сервер, в который загружаются сервлеты. Контейнер сервлетов управляет их выполнением и отдает сформированные страницы Web-серверу или же может сам являться Web-сервером, интегрироваться в другие приложения. Наиболее популярными контейнерами являются *Apache Tomcat* и *Jetty*.

Пример Java сервлета, обрабатывающего запрос *GET*:

```
    import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;

public class NewServlet extends HttpServlet {

@Override
protected void doGet(HttpServletRequest request, HttpServletResponse
response) throws ServletException, IOException {
  String parameter = request.getParameter("name");

  response.setContentType("text/html;charset=UTF-8");
  PrintWriter out = response.getWriter();
  try {
    out.println("<html>\n" +
                "<head>\n" +
                "<title>Заголовок</title>\n" +
                "</head>\n" +
                "<body>\n" +
                "<p>Параметр: " + parameter + "</p>\n" +
                "</body>\n" +
                "</html>");
  } finally {
    out.close();
  }
}
}
```

4.4.6. Расширение HTML

3 *Сервлет – программный интерфейс, реализуемый на языке Java, позволяющий расширить возможности Web-сервера. Это, своего рода, аналог CGI с расширенными возможностями.*

Помимо составления Web-страниц с помощью скриптов, существуют технологии, позволяющие расширять HTML на серверной стороне, то есть встраивать программный код непосредственно в разметку.

Встроенный в HTML разметку PHP код ничем не отличается от скрипта:

```
<html>
  <head>
    <title>Заголовок</title>
  </head>
  <body>
    <?php
        echo "Любой PHP код может быть выполнен в теге <?php ?>";
    ?>
  </body>
</html>
```

Кроме PHP, существуют *JSP* (*Java Server Pages*) для встраивания Java, и *ASP* (*Active Server Pages*) от Microsoft, распространяющийся с *.NET* или его свободной реализацией *.NET-Mono*. Для языков без подобной возможности обычно используются шаблонизаторы – библиотеки отделяющие код отвечающий за логику работы приложения, от разметки, они ничем не уступают подобному подходу.

4.4.7. Сравнение языков программирования для Web-разработки

Здесь кратко рассматриваются способы взаимодействия различных языков программирования с Web-сервером, но не все из них одинаково хороши для этой задачи. Сравнение возможностей, достоинств и недостатков использования различных языков программирования для организации взаимодействия с Web-сервером приведено в таблице 4.1.

Таблица 4.1. Языки программирования для Web-разработки

язык программирования	***плюсы***	***минусы***
C/C++ – мало подходящие языки в силу своей низкоуровневости.	Высокая производительность.	Сложность и низкая скорость разработки. Отсутствие высокоуровневых интерфейсов для работы с Web-сервером.
Java – применяется в крупных, обычно корпоративных, сетях в силу своей направленности на них.	Идеально подходит для крупных проектов из-за организации кода. Относительно высокая производительность.	Сложность и низкая скорость разработки. Высокая стоимость обслуживания Высокий уровень потребления памяти.
PHP – создан для Web-разработки; имеет некоторые архитектурные проблемы.	Легкость и скорость разработки. Огромное количество документации, *фреймворков* и библиотек. Доступен на большинстве хостингов.	Относительно низкая производительность и высокое потребление памяти. Архитектурные ошибки в самом языке.

Perl – мало используется в Web.	Легкость и скорость разработки. Реже доступен на хостингах, чем PHP и Python	Устаревшие модули, документация. Небольшое сообщество и слабая поддержка.
язык программирования	***плюсы***	***минусы***
Python – простой язык с высокой скоростью разработки	Легкость и скорость разработки Огромное количество документации, библиотек и фреймворков. Большое сообщество разработчиков Доступен на многих хостингах	Доступность на хостингах ниже, чем у PHP.
Ruby	*Легкость и скорость разработки.* *Имеющиеся фреймворки (Ruby on Rails, Sinatra) – одни из лучших.*	*Относительно небольшое количество информации, библиотек и фреймворков.* *Низкая доступность на хостингах.*

4.4.8. Фреймворки

Довольно часто при разработке Web-сайтов не пользуются рассмотренными выше технологиями *напрямую*, а используют высокоуровневые библиотеки – *фреймворки*[4], поверх которых строится приложение. Они упрощают многие операции, в том числе размещение на Web-сервере и могут сильно различаться в зависимости от назначения. Существуют фреймворки как для разработки простых Web-сайтов, отвечающие только за размещение или за организацию кода Web-сайта, так и крупные фреймворки, позволяющие создавать комплексные Web-сайты, работающие с пользователями (функции регистрации, авторизации), базами данных и т.д.

Некоторые фреймворки являются агрегацией отдельных библиотек, другие предоставляют лишь функциональность одного типа. Заинтересованным в Web-разработке следует определиться с языком программирования и рассмотреть все возможные варианты, чтобы выбрать наиболее подходящий из них. Ниже приводится таблица «полноценных» (предоставляющих большую часть необходимых функций) популярных фреймворков для различных языков программирования, популярных среди профессиональных Web-разработчиков.

Таблица 4.2. Фреймворки для языков программирования

язык	***комплексные фреймворки***	***простые фреймворки***
Python	Django; TurboGears.	Flask; Web.py; bottle.py.
PHP	Symfony (Drupal); Yii Framework; CakePHP; Codeigniter; Zend Framework; Laravel.	Fat Free Framework; Silex.
Perl	Catalyst.	

4 *Фреймворк (framework – каркас, структура, англ.) – структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта.*

Ruby	Ruby on Rails.	Sinatra.
Java	Spring.	Play.