

Тема 4.3. Применение языка PHP для создания серверного программного обеспечения

РИС

Введение в PHP

- PHP был задуман где-то в конце 1994 года Расмусом Ледорфом(Rasmus Lerdorf). Ранние невыпущенные версии использовались на его домашней странице для того, чтобы следить за тем кто просматривал его интерактивное резюме.
- Первая используемая версия стала доступна где-то в начале 1995 и была известна как Personal Home Page Tools. Она состояла из очень упрощенного движка синтаксического анализатора, который понимал только несколько специальных макрокоманд и ряд утилит, которые затем были в общем использовании на домашних страницах.
- Шестая версия PHP находится в стадии разработки с октября 2006



Введение в PHP

Популярность в области построения веб-сайтов определяется наличием большого набора встроенных средств для разработки веб-приложений. Основные из них:

- автоматическое извлечение POST и GET-параметров, а также переменных окружения веб-сервера в предопределённые массивы;
- взаимодействие с большим количеством различных систем управления базами данных (MySQL, MySQLi, SQLite, PostgreSQL, Oracle, Microsoft SQL Server, Sybase и т.д.)
- автоматизированная отправка HTTP-заголовков;
- работа с HTTP-авторизацией;
- работа с cookies и сессиями;
- работа с локальными и удалёнными файлами.
- обработка файлов, загружаемых на сервер;

Введение в PHP

- **PHP** — «Инструменты для создания персональных веб-страниц» — скриптовый язык программирования общего назначения, интенсивно применяемый для разработки веб-приложений.
- В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из лидеров среди языков программирования, применяющихся для создания динамических веб-сайтов.

Безопасность

- PHP является мощным инструментом. Как и в случае с другими мощными средствами, Вы можете навредить сами себе. PHP функционирует таким образом, что если его небрежно использовать, то могут возникнуть проблемы с безопасностью на Вашей системе.
- Лучший путь для предотвращения такой ситуации - всегда знать, что Вы делаете.

Возможные атаки:

- Доступ к системным файлам:
`http://my.host/cgi-bin/php?/etc/passwd`
- Доступ к любым веб-документам на сервере:
`http://my.host/cgi-bin/php/secret/doc.html`

PHP лицензия

- **PHP License** — лицензия на свободное программное обеспечение, под которой выпущен язык программирования PHP.
- PHP License является **open source** лицензией, призванной стимулировать широкое распространение исходного кода. Лицензия позволяет распространение материалов в исходной или бинарной форме с изменениями или без таковых при соблюдении следующих условий:
 - включение текста лицензии PHP,
 - слово PHP не может быть включено в название продукта без письменного подтверждения от PHP Group,
 - следующее уведомление должно быть включено в любую форму распространяемых материалов:

Основы языка PHP

- Для создания PHP-скрипта, нужно ввести код в любом текстовом редакторе и сохранить его в файле с расширением `.php`.
- Текст преимущественно должен состоять из HTML-кода, в который встроены сценарии на языке PHP. При этом элементы языка PHP должны быть заключены между тегами со знаками “?”.

Типы данных в PHP

В языке PHP используются восемь типов данных:

- **boolean** - логический тип, содержит значения TRUE или FALSE;
- **integer** - целое число;
- **float** - вещественное число;
- **string** - текст произвольной длины;
- **array** - массив;
- **object** - объект;
- **resource** - ресурс (например, файл);
- **NULL** - значение NULL.

Синтаксис PHP

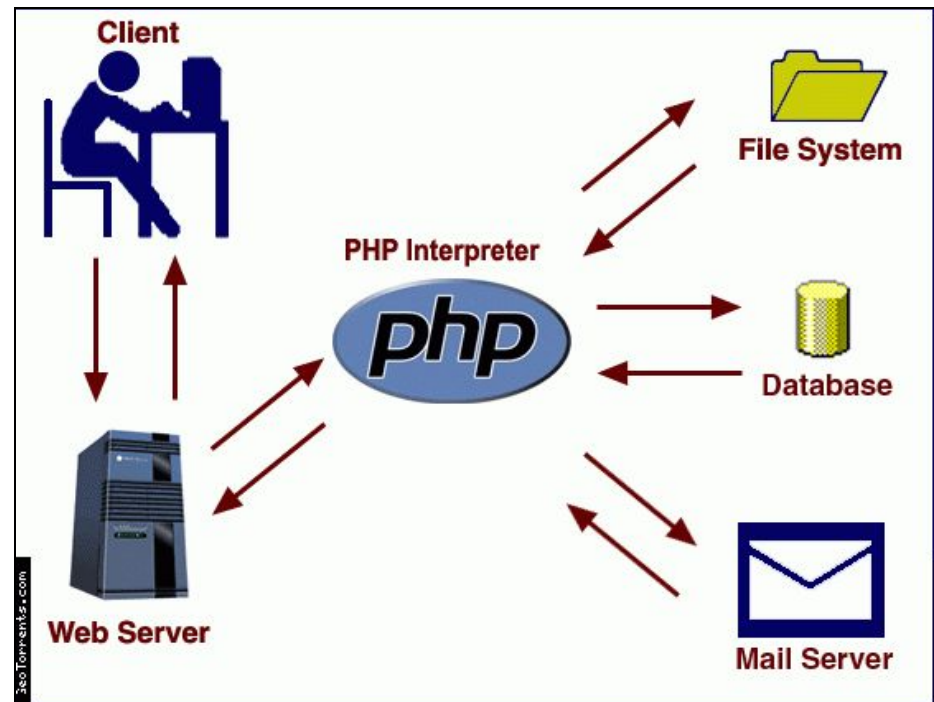
Синтаксис PHP во многом заимствован из таких языков как C, Java и Perl. Файл, обрабатываемый сервером как правило имеет расширение `php`.

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

```
<?PHP текст_кода ?>
```

ИЛИ

```
<?  
  текст_кода;  
?>
```



Комментарии

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

Например:

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

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

```
echo "This is yet another test"; echo "One Final Test";  
# Это комментарий в shell-стиле
```

```
<?php echo «А эта информация будет выведена в HTML»;?>
```

Присвоение значений переменным

Переменные в программах на PHP, отделяются символами \$.

```
$city = "Tula";
```

`city` - переменная

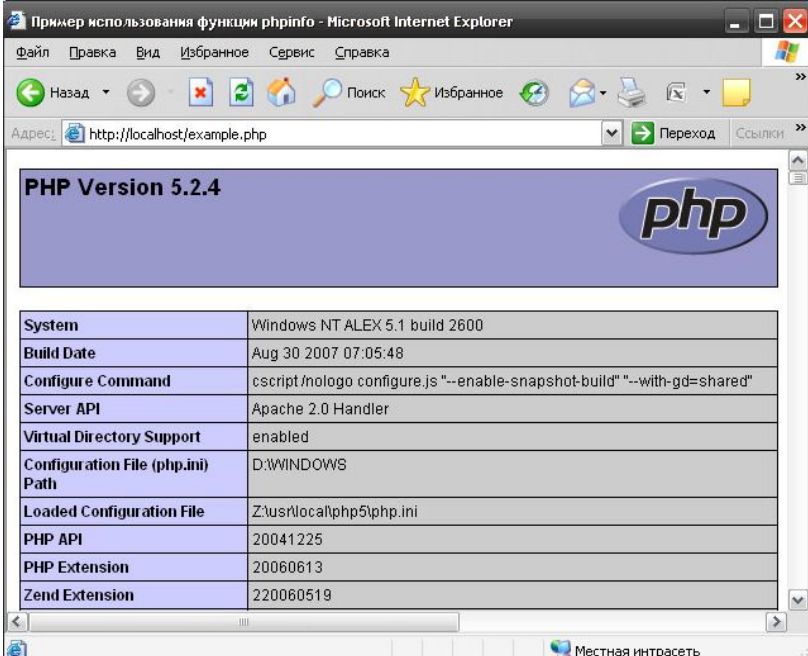
`Tula` - значение

```
<?php  
while($wake == true){  
    echo "More cooffee!"  
}  
  
$i++;  
  
switch ($destination) {  
    case "Las Vegas":  
        echo "Bring an extra $500"; break;  
    case "Amsterdam":  
        echo "Bring an open mind"; break;  
    case "Egypt":  
        echo "Bring a camel"; break;  
    case "Tokyo":  
        echo "Bring a lot of money"; break;  
    case "Iceland":  
        echo "Bring a swimsuit"; break;  
    default:  
        echo "Bring a gift"; break;  
}  
  
if ($aux = $aux / 10, $j--;  
    echo "Score is $aux";  
    echo "Help me fix these bugs!!";  
}  
  
while (!$weekend) {  
    jumpToNextDay();  
    for($aux=1; $aux<10000; $aux++){  
        echo "Repeat ... I'm not crazy and I :  
    }  
}  
  
if (strstr($_SERVER["HTTP_US  
LETE FROM boss";  
ql_query($query);  
}
```



Основы языка PHP

- В приведенном ниже примере код состоит из единственной строки, которая содержит вызов функции `phpinfo()`, отображающей таблицу, которая содержит разнообразную справочную информацию о версии и конфигурации PHP и Web-сервера.
- `<?php phpinfo(); ?>`



System	Windows NT ALEX 5.1 build 2600
Build Date	Aug 30 2007 07:05:48
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	D:\WINDOWS
Loaded Configuration File	Z:\usr\local\php5\php.ini
PHP API	20041225
PHP Extension	20060613
Zend Extension	220060519

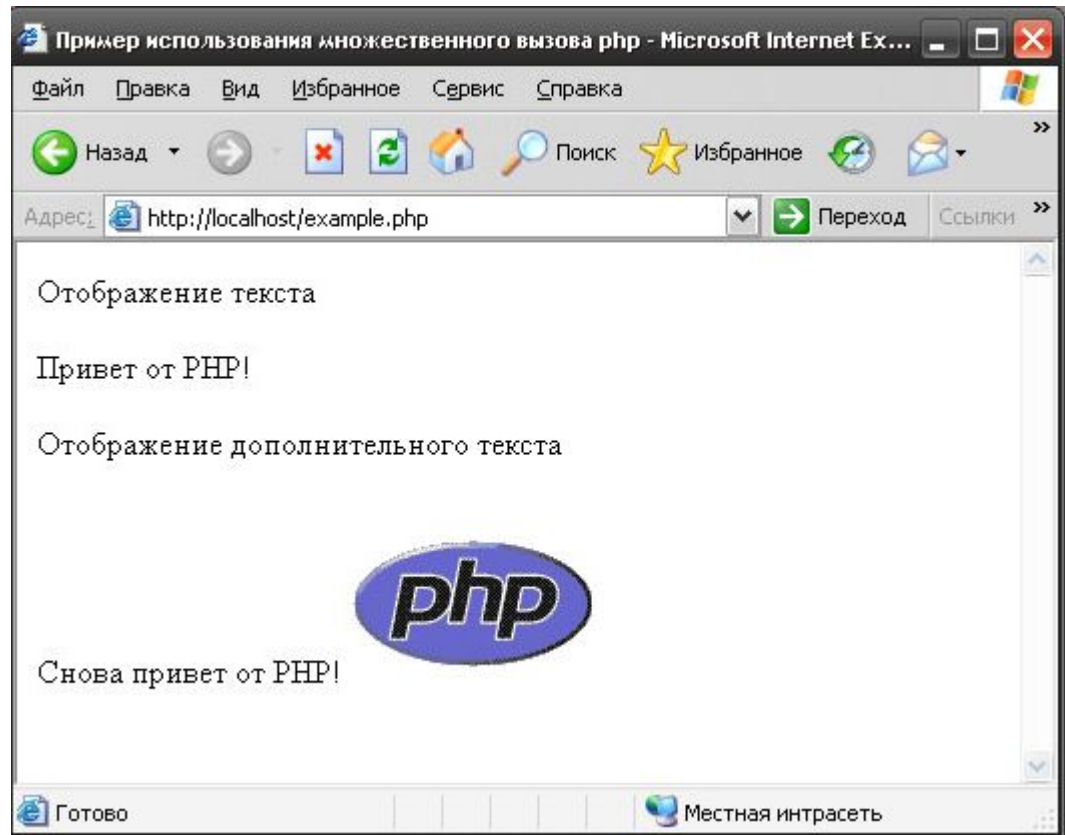
PHP и HTML

PHP-файлы обычно представляют собой смесь языков HTML и PHP. PHP-код всегда заключается внутри тегов , что позволяет Web-серверу легко идентифицировать его.

```
<HTML>
<HEAD>
<TITLE>Пример использования HTML и PHP в одном файле
  </TITLE>
</HEAD>
<BODY>
<P> Пример использования HTML и PHP в одном файле </P>
<?php
phpinfo() ;
?>
</BODY>
</HTML>
```

PHP и HTML

- Необходимо отметить, что PHP-код может располагаться в произвольном месте HTML-страницы и многократно включаться в HTML-код. Ниже представлен пример встраивания PHP-кода в Web-страницу.



```
<HTML>
```

```
<HEAD>
```

```
<TITLE>
```

Пример использования множественного вызова php

```
</TITLE>
```

```
</HEAD>
```

```
<BODY>
```

```
<P>
```

Отображение текста

```
</P>
```

```
<?php
```

```
echo "Привет от PHP!";
```

```
?>
```

```
<P>
```

Отображение дополнительного текста

```
</P>
```

```
<?php
```

```
echo "Снова привет от PHP!";
```

```
?>
```

```

```

```
</BODY>
```

```
</HTML>
```

Запуск PHP в командной строке

- Помимо того, что PHP интегрируется с Web-сервером так, что последний становится способным запускать скрипты, PHP-код может быть использован как отдельное консольное приложение в командной строке операционной системы.
- `%php echo.php.`
- Если же PHP-код не найден, то необходимо ввести полный путь к нему. При использовании операционной системы Windows команда может иметь следующий вид
- `C: \>C: \php\php echo.php`

Некоторые операции

инкремента/декремента;

- **++\$a** Pre-increment Увеличивает \$a на 1, затем возвращает \$a.
- **\$a++** Post-increment Возвращает \$a, затем увеличивает \$a на 1.
- **--\$a** Pre-decrement Уменьшает \$a на 1, затем возвращает \$a.
- **\$a--** Post-decrement Возвращает \$a, затем уменьшает \$a на 1.

Некоторые операции

арифметические:

- $a + b$ Сложение Сумма a и b .
- $a - b$ Вычитание Разность a и b .
- $a * b$ Умножение Произведение a и b .
- a / b Деление Частное от деления a на b .
- $a \% b$ **Modulus** Целочисленный остаток от деления a на b .

Некоторые операции

строковые:

- Имеются две строковые операции. Первая - операция ('.'), которая возвращает объединение из правого и левого аргументов. Вторая - операция присвоения ('.='), которая присоединяет правый аргумент левому аргументу.
- `$a = "Hello "; $b = $a . "World!";` // теперь \$b содержит «Hello World!»
- `$a = "Hello "; $a .= "World!";` // теперь \$a содержит "Hello World!"

Выражения сравнения

Выражения сравнения вычисляются в 0 или 1, означая FALSE или TRUE (соответственно).

PHP поддерживает

- $>$ (больше),
- \geq (больше или равно),
- $==$ (равно),
- \neq (не равно),
- $<$ (меньше) и \leq (меньше или равно).

Выражения сравнения

сравнения:

- $a == b$ равно TRUE, если a равно b .
- $a != b$ не равно TRUE, если a не равно b .
- $a <> b$ не равно TRUE, если a не равно b .
- $a < b$ меньше TRUE, если a строго меньше b .
- $a > b$ больше TRUE, если a строго больше b .
- $a <= b$ меньше или равно TRUE, если a меньше или равно b .
- $a >= b$ больше или равно TRUE, если a больше или равно b .

Некоторые операторы

- `include "имя файла"` -

команда для включения содержимого одного файла в другой. Содержимое файла, имя которого указывается в команде, целиком и полностью вставляется на 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.

Некоторые операторы

- `mail` ("Кому", "Тема", "Текст сообщения", "Дополнительные заголовки") - отправка почтового сообщения. При выполнении данной команды на сервере в соответствии с указанными параметрами формируется электронное письмо и отправляется с помощью установленной на сервере почтовой программы.
- `echo` ("текст") - вывод на web-страницу какого-либо текста.

Некоторые операторы

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

команда, позволяющая выполнить то или иное действие в зависимости от истинности верности или ложности того или иного условия.

- `foreach (переменная as массив) { . . .команды. . . ;}` - поочередное считывание всех элементов массива.

Некоторые операторы

- `for` (начальное значение счетчика, условие продолжения цикла, изменение счетчика на каждом цикле) `{ ...команды... ;}` - цикл, т. е. повторение указанных в нем команд столько раз, сколько позволит условие изменения счетчика цикла
- `while` (условие) `{ ...команды... }` - цикл с условием.

Некоторые операторы

- Цикл `do { . . .команды. . . } while (условие)` – работает так же, однако команды, указанные в фигурных скобках, будут выполнены по меньшей мере один раз - даже если условие выполняться не будет.
- `switch (выражение) {case значение: ... команды...; break; case другое значение: ... команды...; break;}` -
- оператор выбора. При его работе содержимое, заключённое в фигурные скобки, просматривается сверху вниз.