PHP

Курсова работа на Тодор Арнаудов, 0326037

Какво е PHP?

PHP е интерпретиран скриптов език за програмиране, който е създаден да работи на сървърната страна на клиент-сървърни приложения в уеб. Видимият резултат от действието на кода на PHP най-често са страници на HTML.

Началото на езика е поставено през 1994 г., някъде в ранните години на WWW, от Рамус Лерфорд, кодирал зад съкращението звучащите комично сега думи Personal Home Page. Донякъде поради по-късната неадекватност на тези думи на фона на разрасналите се приложения на езика, донякъде заради традиция в именуване на приложенията с лиценз с отворен код под GNU, каквото е PHP, по-късно името се променя на PHP Hypertext Preprocessor. Това име много по-точно изразява функцията на системата PHP, защото кодът на PHP може да се вгражда между HTML, и PHP се използва за динамично генериране на HTML-страници.

Системата PHP е с отворен код, допълнения за нея пишат хиляди програмисти и е една от най-използваните за създаване на динамични уеб сайтове. З а основната й разработка се грижи фирмата Zend, като последната версия на PHP към момента е 5.2.0. [ноември 2006 г.]

Езикът е характерен с много версии, които не винаги са съвместими отдолу-нагоре дори във важни моменти от синтаксиса и вградените функции (като напр. таговете за отваряне и затваряне на блок с код на PHP в хтмл), което става причина за проблеми при пренасянето на системи към по-горни версии, което пък от своя страна забавя разпространението на новите версии.

PHP е характерен и с огромното количество вградени функции и разширения (extensions), които се разпространяват безплатно.

Макар че PHP е най-удобен за създаване на динамични уебстраници – форуми, блогове, електорнни магазини – и за обработка на текст, поради вградените възможности за работа с регулярни изрази и множеството готови функции за обработка на низове, с негови разширения могат да се генерират и обработват и изображения (gd), да се създават клипове на Флаш (ming), да се прекодират видеофайлове.

Домашният сайт на PHP: http://PHP.net съдържа подробно описание на действието на всичко, що се отнася до PHP, и се обогатява с примери и съвети и от самите потребители.

MySQL

PHP е тясно свързан със СУБД MySQL, която е с отворен код и е безплатна за ползване в некомерсиални приложения.

Характеристики на PHP като език за програмиране

Накратко, PHP е интерпретиран език, слабо типизиран; все още е преобладаващо процедурен, но от версия 5 на него може да се пише и се насърчава да се пише обектноориентирано.

Слабата типизация се изразява в това, че променливите не се дефинират с изричен тип, а типът се разпознава по време на изпълнение и може да се променя без изрично преобразуване.

Бързо запознаване с особености на синтаксиса на PHP чрез примери

Операторите за блок в PHP са като на Си: къдрави скоби {}; знакът за предаване на параметър по референция е амперсанд & също като в Си и Си++, командите се разделят с ";". Условните оператори **if else**, циклите **for, while**, многовариантния избор **switch** приличат на съответните в Си; операторът за указване на клетка от масив е също като на Си **[]**. Много от вградените функции в PHP имат идентични имена с функции от библиотеки на Си, напр. sprintf. От пръв поглед върху кода на PHP се вижда прилика и с друг език, предшественик на PHP, който също е подобен и на Си – PERL.

Какви характерни неща забелязваме в PHP?

-- Променливите винаги се означават с **$** и нямат изричен тип, макар че PHP работи с булеви стойности, числови стойности и низове. Типът на данните се присвоява по време на изпълнение – веднъж една стойност може да се възприема като низ, друг път като число, без да е нужно изрично преобразуване, чрез т.нар. тип **"mixed"**.

-- Масивите могат да съдържат непоредни индекси, да се уголемяват динамично без извикване на специални функции, и да съдържат елементи от разнороден тип.

- Асоциативните масиви (хеш таблици) - масиви които могат да използват низове като индекси - са вградени в езика.

```php
<?PHP        // таг за начало на блок с код

$var = ”Something”;
echo ”Print $var”; a Print Something
echo ‘Print $var’;  a Print $var
echo ”<font size=4 color=”blue”><br>” // Генериране на тагове
на хтмл
$var = 123;
$var = 456.789;
$arr = array();
$arr[0] = 1;
$arr[1]=”низ”;
$arr[5] = ”масиви с дупки”;

?>    //таг за край на блок с код

<div style=”position: relative; right: -80px;”> // вмъкване на
директен ХТМЛ


<?PHP

function doo($byValue,  &$byReference)
{
  $newValue = 10;
  $byValue = $newValue;
  $byReference = $newValue;
}

$A = 1;
$B = ”Dog”;
doo($A, $B);
echo ”<p>A = $A, B = $B”;           // A==1, B==10

/* Променливата, предадена по стойност, не е променена извън
функцията, защото във функцията се създава ново копие.
Стойността на променливата, предадена по референция, се
променя, и вече не е низ, а число. */

$arr2 = array(1, 5, ”word”, ”house”, ”computer”, ”man”, 124.34
6);
$assoc = array( ‘key1’=> ‘value1’, ‘key2’ => ‘value2’, ‘кола’
=> ‘car’);
```

```
foreach ( $arr2 as $value)
  echo ”<p>Извлича елемент от масива и прави нещо със стойност
та му”;

foreach ( $arr2 as $key => $value)
  echo ”<p> Извлича елемент от асоциативен масив и прави нещо
със стойността му”;

$i = 2;

switch ($i):
case 0: echo ”I = 0”;
break;
case 1: echo ”I = 1”;
break;
default: echo ”I is biiiig!”;
endswitch;

$boolyA = true;
$boolyB = true;

if ($booly===true) {      // ===  е оператор за сравняване на с
тойности, които са изрично булеви

   print "booly is true<BR>\n";
} elseif ($b) {
                      echo ”BBBB…”;
             }

/*
  Лесно можете да оцветите код PHP чрез функциите:
  mixed highlight_file ( string filename [, bool return] )
  mixed highlight_string ( string str [, bool return] )
*/
```

Принцип на работа на клиент-сървърни уеб приложения с PHP

Начини за предаване на стойности между скриптове

Динамичните сайтове на PHP представляват множество от скриптове (файлове с код на PHP), които генерират XTML-страници. За предаване на данни между различните скриптове се използват няколко начина:

1. Чрез глобални променливи
2. Чрез масива POST
3. Чрез масива GET
4. Чрез база данни

1. Предаване на данни чрез глобални променливи

```
<?PHP session_start();
```

/* необходимо е да сме включили сесия, като функцията session_start трябва да е записана непосредствено след тага за начало на блок с код - без излишни интервали или празни редов, - в противен случай ще се получи съобщение за грешка
*/

```
$_SESSION['global_variable_name']= 123;
$_SESSION['global_variable_name2'] = $variable
```

2. Предаване на данни чрез формуляри и масива POST

За да прочетем стойности, въведени от потребители в даден скрипт, и да ги предадем на следващия, използваме формуляри.

```
<form name="login" action="doLogin.PHP" method="post">
<font size=4>
Username: <input type="text" name="username">
<p>
Password: <input type="password" name="password" method="post">
<p>
<input type="submit" action="doLogin.PHP" value="Log In">
</form>
</center>
```

Така създаваме формуляр, съдържащ две едноредови текстови полета, едното предназначено за парола (знаците в него излизат като звездички), и бутон за изпращане с надпис "Log In". При натискане на бутона за изпращане, се извиква скриптът doLogin.PHP, и данните скрито се предават на системата PHP, и няма явно ограничение за дължината им. Данните от текстовите полета могат да се прочетат във втория скрипт от служебния масив **$_POST.**

```
      if (   ( !isset($_POST['username'])) || (!
isset($_POST['password']) ) )
   {
    echo '<br>You have to enter username and password!';
   }
```

Служебният масив $_POST['username'] съдържа стойността на съответния входен контрол.

Функцията isset($var) служи за проверка дали променливата е била поне инициализирана с някаква стойност. Добре е да се правят такива проверки и да не се допускат скриптовете да "гърмят" с редове с грешки.

! е оператор за лог. отрицание, както в Си.

3. Предаване на данни в адреса и чрез масива GET

При този начин, имената на параметрите и стойностите им се предават в адреса. Това е удобно в някои случаи – напр. адресът може да се запомни и да се извика по-късно без потребителят да пише отново във формуляри; можем и да генерираме такива адреси чрез код, което също е много удобно.
Форматът е следният:

http://www.something.com/script.PHP?parameter1=Value1&parameter2=123

Пример за генериране на връзка, която предава стойност към друг скрипт, който записва въведените данни в БД или извършва друга операция, зададена от входните данни.

```
echo "<a href=\"edit.PHP?id=$postID\">[edit]</a>";
```

Максималната дължина на стойност трябва да е <= 100 знака, което подсказва, че методът не е подходящ за предаване на текст, въведен например във форум или блог. Освен това, очевидно след като стойността на параметрите се вижда в адреса, този метод е неподходящ за предаване на пароли.

4. Предаване на данни чрез база данни

Това става като записваме данните трайно в БД MySQL, която е неизменен спътник на PHP; съответно като извличаме от БД нужната ни информация чрез подходящи заявки. Следният код илюстрира четене от БД:

```
$db = mysql_connect('localhost', 'username', 'password');
mysql_select_db("DB_NAME", $db);         //избира БД
$query  = "select * from data order by datetime DESC";
//подготвя заявката
$result = mysql_query($query, $db);    //изпълнява заявката
$num_rows = mysql_num_rows($result);     //връща брой колони в
резултата
for($i=0; $i<$num_rows; $i++)
{
$row = mysql_fetch_array($result); //асоциат. масив с резултат
а от                                              заявката
  $postID = $row['postid'];
  $authorID = $row['authorID'];
  $posttime = $row['datetime'];
 //....
}
```

...

Тази работа имаше за цел само да даде пример, за повече информация, препоръчваме:

http://google.com

http://php.net
http://zend.com
http://wampserver.com
http://www.w3schools.com/sql/
http://www.w3schools.com/php/
http://www.w3schools.com/html/
http://www.w3schools.com/css/

11.2006 г.