

PHP – PHP Hypertext Preprocessor

Igor Wojnicki

Katedra Informatyki Stosowanej
Akademia Górniczo-Hutnicza w Krakowie

4 grudnia 2013

\$Id: php0.tex,v 1.4 2013/10/27 09:32:42 wojnicki Exp \$

Spis Treści

- 1 PHP
 - Opis Języka

I. Wojnicki, JiTW

Literatura

- White E., Eisenhamer, J.D.: PHP 5 w praktyce. Helion, Gliwice 2007.
- Lis M.: PHP 5: praktyczny kurs. Helion, Gliwice 2006.

I. Wojnicki, JitW

Spis Treści

- 1 PHP
 - Opis Języka

I. Wojnicki, JiTW

Język I

- PHP (angielski akronim rekurencyjny, którego rozwinięcie to PHP Hypertext Preprocessor), pierwotnie nazwany Personal Home Page - skryptowy język programowania, służący przede wszystkim do tworzenia dynamicznych stron WWW i wykonywany w tym przypadku po stronie serwera, z możliwością zagnieżdżania w HTML (bądź XHTML).
- PHP jest podobny w założeniach do dużo starszego mechanizmu SSI (Server Side Includes), jednak jest w stosunku do SSI nieporównanie bardziej rozbudowany.
- Udostępniany jest na zasadach licencji open-source. Jego składnia bazuje na językach C, Java i Perl.

Język II

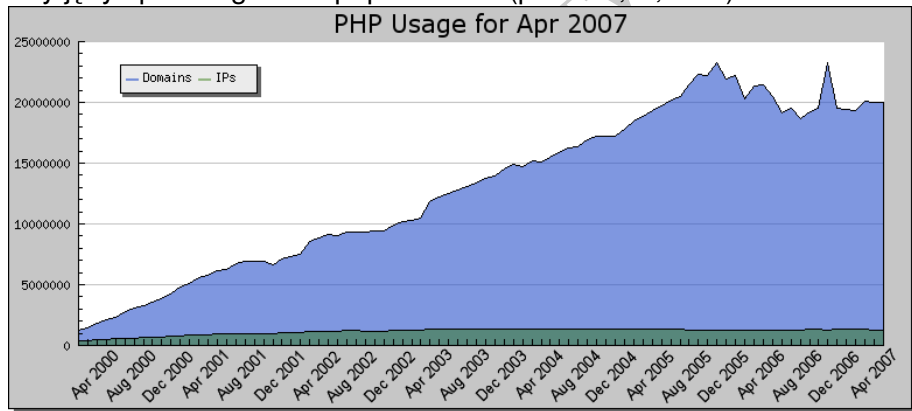
- PHP może być używane nie tylko do tworzenia stron WWW. Jego modułowa budowa pozwala także na programowanie aplikacji z interfejsem graficznym (rozszerzenie PHP-GTK), a także na wykonywanie z linii poleceń (podobnie jak Perl i Python).
- PHP umożliwia także współpracę z wieloma systemami relacyjnych baz danych (np. MySQL, Oracle, PostgreSQL, SQLite) oraz korzystanie z alternatywnych sposobów przechowywania danych - plików tekstowych i XML-owych. Może być uruchamiane na większości systemów operacyjnych (uwzględniając najpopularniejsze) oraz serwerów sieciowych.
- PHP w środowisku Linux wraz z serwerem Apache i silnikiem baz danych MySQL stanowi popularną platformę serwerową, tzw. LAMP.
- Od wersji 5 udostępnia wydajne mechanizmy obiektowe.

Popularność PHP

Kwiecień 2007

PHP: 20,917,850 domen, 1,224,183 adresów.

4-ty język pod względem popularności (po Java, C, C++)¹.



¹<http://langpop.com/>

Charakterystyka I

- Interpreter.
- Zastosowania Internetowe.
- Interpreter Stand-alone.

Możliwości:

- TCP/IP, DNS.
- http, https, ftp, gopher, telnet, ldap, irc, snmp, ssh.
- eMail.
- IMAP (Internet Message Access Protocol, Internet Mail Access Protocol), POP3 (Post Office Protocol), NNTP (Network News Transfer Protocol).
- dBase, DBM (BerkeleyDB, GDBM), PostgreSQL, Oracle, MySQL, SQL Server, FrontBase, ODBC, Sybase, SQLite, InterBase, Firebird, IBM DB2, Informix, mSQL, Paradox.

Charakterystyka II

- Niskopoziomowe I/O.
- Expect: Interakcja z zewnętrznymi aplikacjami poprzez stdin/stdout/stderr.
- gnuPG.
- ID3, oggvorbis..
- Transofrmacje: PNG, JPEG, GIF, TIFF.
- Exif.
- Kerberos.
- Uruchamianie metod obiektów Java.
- Przetwarzanie XML, XSLT.
- Kompresja: bzip2, lzf, gzip, zip.
- Mimetype (jaki typ pliku?).
- PDF, Postscript.

Charakterystyka III

- Wyrażenia Regularne.
- Konwersja znaków diakrytycznych: GNU Recode.
- Obsługa Sesji za pomocą cookies.
- Obliczenia Statystyczne.
- ncurses: interakcja w środowisku tekstowym.

Osadzanie I

- Instrukcje zakończone średnikiem (;) [1]

- `<?php echo("Tralala Bum!\n"); ?>`

- ```
<?php
if (wyrazenie-logiczne) {
 ?>
prawda
 <?php
} else {
 ?>
fałsz
 <?php
}
?>
```

# Komentarze I

- jak dla języka C: `/* ... */`
- jak dla języka C++ (do końca linii): `//`
- jak dla powłoki sh (do końca linii): `#`

I. Wojnicki, JitW

# Typy Danych I

PHP obsługuje osiem typów prostych.

- boolowski (boolean): True/False

```
$foo = True;
```

- liczba całkowita (integer)

mogą być zapisane w notacji dziesiętnej, szesnastkowej (0x) lub ósemkowej (0),

- liczba zmiennoprzecinkowa (float)

```
$a = 1.234; $a = 1.2e3; $a = 7E-10;
```

# Typy Danych II

- łańcuch znaków (string), ciąg bajtów

```
echo 'przykładowy tekst';
```

- podwójne cudzysłowy, interpretacje większej ilości sekwencji cytowania dla znaków specjalnych.

```
echo "tab \t nowa\n";
```

- *heredoc*

```
$str = <<<EOD
```

```
Przykład łańcucha znaków,
zajmującego kilka linii,
zapisanego w składni heredoc.
EOD;
```

# Typy Danych III

- tablica (array)

```
$tablica=array(1,2,3,'cos',5);
echo $tablica[0];
```

```
$inna=array(
 1 => 'jeden',
 5 => 'piec',
 'trzy' => 3
);
echo $inna[5];
echo $inna['trzy'];
```

- obiekt (object)

# Typy Danych IV

- identyfikator zasobów (resource) jest specjalną zmienną, przechowującą odnośnik do zewnętrznego źródła zasobów. Identyfikatory zasobów są tworzone i wykorzystywane przez specjalne funkcje (zobacz suplement z listą tych funkcji i powiązanych z tym typów zasobów w dokumentacji PHP). Uwaga: Identyfikator zasobów został wprowadzony w PHP 4.
- NULL wartość specjalna oznacza, że zmienna nie przechowuje żadnej wartości. Uwaga: typ NULL został również wprowadzony w PHP 4

PHP ustala typ danych na podstawie kontekstu.



# Zmienne I

- Każdą zmienną w PHP zapisuje się, poprzedzając jej nazwę znakiem dolara \$.
- Dostępne są funkcje które mogą sprawdzić typ danych:  
`is_array()`, `is_int()`, `is_float()` ... itp.
- oraz istnienie samej zmiennej i jej wartości: `isset()`, `isnull()`  
– uwaga na różnicę, zobacz: `unset()`!
- Wielkość liter w nazwie zmiennej jest rozróżniana.

# Zmienne II

- Referencje: & (od PHP4)

```
<?php
$foo = 'PHP';
$bar = &$foo;
$bar = "To jest $bar";

echo $bar;
echo $foo;
?>
```

- Zasięg zmiennej ogranicza blok, w którym jest zdefiniowana.
- `static` zmienne statyczne (w ramach funkcji!!!)
- Zmienne jako nazwy zmiennych `${$var}=5;`

# Zmienne Globalne I

- `$_GLOBALS` referencje do zmiennych globalnych, nazwa zmiennej = klucz
  - `$_SERVER` tworzone przez serwer http
  - `$_GET` `$_POST` zmienne dostarczone do skryptu metodą GET lub POST
  - `$_COOKIE` zmienne dostarczone przez mechanizm COOKIES
  - `$_REQUEST` zawiera informacje z get/post/cookie
  - `$_FILES` zmienne dostarczone przez przesłanie plików
  - `$_ENV` zmienne środowiskowe, kontekst uruchomienia skryptu
- 1 Dla wersji poniżej 4.1 obowiązywały nazwy zmiennych globalnych w formacie: `$HTTP_*_VARS`.
  - 2 Dla PHP3 dane przekazywanem przez POST/GET były zmiennymi globalnymi.

# Stałe I

Stała jest identyfikatorem (nazwą) dla prostej wartości. Jak sama nazwa wskazuje, wartość ta nie może się zmieniać podczas działania skryptu (poza wyjątkami: `__FILE__` i `__LINE__`). Domyślnie, przy stałych uwzględniana jest wielkość liter. Przyjęto, że stałe są pisane **dużymi literami**.

```
define('WEB_MASTER', 'nczbik@cyf-kr.edu.pl');
echo WEB_MASTER;
```

# Operatory I

- Arytmetyczne - analogiczne do C: + - / \* %
- Przypisania: = += -=
- Bitowe: & | ~ ^ << >>
- Porównania: == === != <> !== < > = >
- Warunkowe: (expr) ? (expr1) : (expr2);
- Wykonania: ` `
- Inkrementacji/Dekrementacji: ++ --
- Logiczne: and or xor ! && ||
- Konkatenacja łańcuchów znaków: . .=
- Łączenie tablic: +

# Struktury Sterujące I

- if elseif else

```
if (wyrażenie) { //1 postać
 // jeśli wyrażenie jest prawdziwe
}
if (wyrażenie) { //2 postać
 // jeśli wyrażenie jest prawdziwe
}
else {
 // jeśli wyrażenie jest jest fałszywe
}
```

# Struktury Sterujące II

```
if (wyrażenie1) { //3 postać
 // jeśli wyrażenie1 jest prawdziwe
}
elseif (wyrażenie2) {
 // jeśli wyrażenie2 jest prawdziwe
}
else {
 //wykonywane jest jeśli wszystkie
 //wyrażenia są fałszywe
}
```

# Struktury Sterujące III

- switch, analogicznie do C

```
switch (zmienna) {
 case WARTOSC_1:
 //instrukcje do wykonania...
 break;
 case WARTOSC_2:
 break;
 case WARTOSC_n:
 break;
}
```



# Struktury Sterujące IV

- for

```
for (INICJACJA; WARUNEK; OPERACJA) {
 //blok instrukcji
}
```

- while

```
while (WARUNEK) {
 // blok instrukcji
}
```

- do

```
do {
 // blok instrukcji
} while (WARUNEK);
```

- break, continue, analogicznie jak C

# Struktury Sterujące V

- foreach, iteracja na elementach tablicy (od PHP4)

```
foreach(array_expression as $value)
 statement
```

```
foreach(array_expression as $key => $value)
 statement
```

# Struktury Sterujące VI

- funkcje

```
function nazwa ($arg1, $arg2) {
 //wiersze kodu
 return $my_var; // opcjonalne
}
```

- `require`, `include` – **wczytanie skryptu z innego pliku, w przypadku niepowodzenia `include` generuje ostrzeżenie, `require` błąd krytyczny.**

```
include 'vars.php';
require 'A.php';
```

# Funkcje I

- Funkcji nie można przeciążać... bo zbyt ciężkie się zepsują.
- Przekazywanie argumentów przez wartość.
- Przekazywanie argumentów przez referencje:

```
function dodaj_cos_extra(&$string)
{
 $string .= 'i coś extra.';
}
```

- Wartości domyślne argumentów.

```
function robkawe ($typ = "cappucino")
{
 return "Robię kubek $typ.\n";
}
```

# Funkcje II

- Zwracanie referencji.

```
function &zwrocReferencje(){
 return $jakasref;
}
$nowaref =& zwrocReferencje();
```

# Funkcje III

- Nazwa funkcji jako zmienna.

```
function foo() {
 echo "W foo()
\n";
}
function bar($arg = '') {
 echo "W bar(); argumentem
 jest '$arg'.
\n";
}
$func = 'foo';
$func();
$func = 'bar';
$func('test');
```

# Interakcje z Serwerem WWW I

- Formularze.

```
<form action="foo.php" method="post">
 Name:<input type="text"
 name="username">

 <input type="submit">
</form>
```

zmienna `$_POST["username"]` otrzyma wartość wprowadzoną przez użytkownika.

- Cookies. Analogicznie jak `$_POST/$_GET`, wszystkie cookies są przesłane w tablicy `$_COOKIE`, funkcja `setcookie()` tworzy/modyfikuje wartości cookies (funkcja może być użyta jedynie zanim skrypt cokolwiek wygeneruje na wyjściu).

# Przykładowa Obsługa Formularza I

Dane klienta w polach formularza: DATA\_\*

Zakupione produkty: PROD\_\*

```
<html>
<head>
<meta HTTP-EQUIV="Content-Type"
 CONTENT="text/html;
 charset=iso-8859-2">
<meta http-equiv="Content-Language"
 CONTENT="pl">
<TITLE>podziekowanie</TITLE>
</head>
<body>
<?php
$to = 'kowalski@serwer.poczta.pl';
```



## Przykładowa Obsługa Formularza II

```
$subject = 'Zamowienie';
$body = "Zamowienie:\n\n";
foreach ($_POST as $k => $v) {
 if (!strncmp($k, 'DATA_', 5)) {
 if ($v != '') {
 $n=substr($k, 5);
 $body="$body}{$n}: $v \n";
 }
 }
}
$body="$body\n";
foreach ($_POST as $k => $v) {
 if (!strncmp($k, 'PROD_', 5)) {
 if ($v != '') {
 $n=substr($k, 5);
```

## Przykładowa Obsługa Formularza III

```
 $body="{body}produkt: $n ilosc: $v \n";
 }
}
}
$body="{body}\n
 Razem do zapłaty:$_POST[TOTAL]";
if (mail($to, $subject, $body))
{
?>
 <p>
 Dziękuję za złożenie zamówienia

<?php
}
else {
?>
```

# Przykładowa Obsługa Formularza IV

```
<p>
 Błąd!!! Zamówienie nie zostało wysłane !!!
<?php
 }
?>
</body>
</html>
```

I. Wojnicki, JITW

# Rodzaje Kontroli Dostępu

- HTTP authorization,
- Formularze HTML
- + POST/GET + COOKIES + DBMS.

I. Wojnicki, JitW



*Podręcznik PHP.*

[http://www.php.net/manual/pl/.](http://www.php.net/manual/pl/)

I. Wojnicki, JiTW