

PHP – PHP Hypertext Preprocessor

Igor Wojnicki

Katedra Informatyki Stosowanej
Akademia Górniczo-Hutnicza w Krakowie

- 1 PHP
 - Opis Języka

I. Wojnicki, PHP

Spis Treści

- 1 PHP
 - Opis Języka

I. Wojnicki, PHP

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

Rok 2021.

TIOBE <http://www.tiobe.com/tiobe-index/> **9:** Python, C, Java, C++, C#, VisualBasic, JavaScript, SQL, PHP,

PYPL <http://pypl.github.io/PYPL.html> **6:** Python, Java, Javascript, C#, C/C++, PHP,

IEEE SPECTRUM <https://spectrum.ieee.org/top-programming-languages/> **13:** Python, Java, C, C++, JavaScript, C#, R, Go, HTML, Swift, Arduino, Matlab, PHP.

Charakterystyka I

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

I. Wojnicki, PHP

Charakterystyka II

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.
- Niskopoziomowe I/O (bezpośrednia komunikacja z urządzeniami peryferyjnymi).

Charakterystyka III

- 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.
- Wyrażenia Regularne.

Charakterystyka IV

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

Osadzanie I

- Instrukcje zakończone średnikiem (;) [1]
- w dokumencie HTML przy użyciu znacznika `php`

```
<html>
...
  <?php
    echo("Tralala Bum!\n");
  ?>
...
  <?php
    echo("W innym miejscu w dokumencie!\n");
  ?>
</html>
```

Komentarze I

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

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., jak również `gettype()` zwracające nazwę typu danych,
- oraz istnienie samej zmiennej i jej wartości:
 - `isset()` (istnieje i różne od NULL),
 - `is_null()`,
 - zobacz: `unset()`
- Wielkość liter w nazwie zmiennej jest rozróżniana (**w nazwach funkcji i słowach kluczowych nie!**).

Zmienne II

- Referencje: & (od PHP4)

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

echo $bar; //To jest PHP
echo $foo; //To jest PHP
?>
```

- 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.

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.

Rzutowanie

- (int), (integer) - rzutowanie na int
- (bool), (boolean) - rzutowanie na bool
- (float), (double), (real) - rzutowanie na float
- (string) - rzutowanie na string
- (array) - rzutowanie na array
- (object) - rzutowanie na object
- (unset) - rzutowanie na NULL

```
$a=3.34;  
echo gettype($a); // double  
$b=(int)$a;  
echo gettype($b); // integer  
echo $b;           // 3
```

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**. Mogą być jedynie typów prostych: integer, float, string, boolean.

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

Operatory

- Arytmetyczne - analogiczne do C: + - / * %
- Przypisania: = += -=
- Bitowe: & | ~ ^ << >>
- Porównania: == === != <> !== <= < >= >
 (5===5); (5=== '5'); (5===(int)'5');
 (5=='5x'); (5=== '5x');
- Warunkowe: (expr) ? (expr1) : (expr2);
- Wykonania: ` `
- Inkrementacji/Dekrementacji: ++ --
- Logiczne: and or xor ! && ||
- Konkatenacja łańcuchów znaków: . .=
- Łączenie tablic: +

Operatory

- Arytmetyczne - analogiczne do C: + - / * %
- Przypisania: = += -=
- Bitowe: & | ~ ^ << >>
- Porównania: == === != <> !== <= < >= >
 (5===5); (5=== '5'); (5===(int)'5'); **true false true**
 (5=='5x'); (5=== '5x');
- Warunkowe: (expr) ? (expr1) : (expr2);
- Wykonania: ` `
- Inkrementacji/Dekrementacji: ++ --
- Logiczne: and or xor ! && ||
- Konkatenacja łańcuchów znaków: . .=
- Łączenie tablic: +

Operatory

- Arytmetyczne - analogiczne do C: + - / * %
- Przypisania: = += -=
- Bitowe: & | ~ ^ << >>
- Porównania: == === != <> !== <= < >= >
 (5===5); (5=== '5'); (5===(int)'5'); **true false true**
 (5=='5x'); (5==='5x'); **true false**
- Warunkowe: (expr) ? (expr1) : (expr2);
- Wykonania: ` `
- Inkrementacji/Dekrementacji: ++ --
- Logiczne: and or xor ! && ||
- Konkatenacja łańcuchów znaków: . .=
- Łączenie tablic: +

Struktury Sterujące I

- if elseif else

```
<?php
    if (2*2==4) echo 'prawda';
?>
```

```
<?php
    if (2*2==5) {
        echo 'prawda';
    } else {
        echo 'fałsz';
    }
?>
```

Struktury Sterujące II

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

Struktury Sterujące III

```
if (wyrazenie1) { //3 postać
    // jeśli wyrażenie1 jest prawdziwe
}
elseif (wyrażenie2) {
    // jeśli wyrażenie2 jest prawdziwe
}
else {
    //wykonywane jest jeśli wszystkie
    //wyrażenia są fałszywe
}
```

Struktury Sterujące IV

```
<?php
if (2+2==4) {
    ?>
    <strong>prawda</strong>
    <?php
} else {
    ?>
    <strong>fałsz</strong>
    <?php
}
?>
```

```
<?php
if (2+2==4) {
    echo '<strong>prawda</strong>';
} else {
    echo '<strong>fałsz</strong>';
}
?>
```

Struktury Sterujące V

- switch, analogicznie do C

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

Struktury Sterujące VI

```
$wynik='kareta';

switch ($wynik) {
    case 'kareta':
        echo 'Wow, nieźle!';
        break;
    case 'full':
        echo 'Tylko tyle?';
        break;
    case 'poker':
        echo 'Rozbijesz bank!';
        break;
    default:
        echo 'Nie masz nic ciekawego...';
}
```

Struktury Sterujące VII

- for

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

- while

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

- do

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

- break, continue, analogicznie jak w języku C.

Struktury Sterujące VIII

```
$j=0;
for ($i=1; $i<5; $i++) {
    $j+=$i;
}
$j=0; $i=1;
while ($i<5) {
    $j+=$i;
    $i++;
}
$j=0; $i=1;
do {
    $j+=$i;
    $i++;
} while ($i<5);
```


Struktury Sterujące IX

- foreach, iteracja na elementach tablicy (od PHP4)

```
foreach(tablica as $wartosc)
```

```
...
```

```
foreach(tablica as $klucz => $wartosc)
```

```
...
```

Struktury Sterujące X

```
$a=array(1=>'jeden', 'piec'=>5);  
foreach ($a as $k => $w) {  
    echo $k.' '. $w."\n";  
}
```

I. Wojnicki, PHP

Struktury Sterujące XI

- funkcje

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

- Wielkość liter w nazwie zmiennej jest rozróżniana, **w nazwach funkcji i słowach kluczowych nie!**
- `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.

```
function razydwa($x) {  
    return $x*2;  
}  
echo razydwa(5); // 10
```

```
function r2($x) { $x*=2; return $x; }  
$a=5;  
echo r2($a); // 10  
echo $a; // 5
```

Funkcje II

- Przekazywanie argumentów przez referencje:

```
function ładny(&$string) {  
    $string = 'ładny ' . $string;  
}  
$x='samochód';  
ładny($x);  
echo $x; // ładny samochód
```

Funkcje III

- Wartości domyślne argumentów.

```
function rob($typ = 'cappuccino') {  
    return "Robię kubek $typ.\n";  
}  
echo rob();  
echo rob('herbaty');
```

Funkcje IV

- Zwracanie referencji.

```
function &zwrocReferencje() {  
    static $zmienna=0;  
    $zmienna++;  
    return $zmienna;  
}  
$referencja=&zwrocReferencje();  
$wartosc=zwrocReferencje();  
echo $wartosc;           // 2  
echo $referencja;       // 2
```

Funkcje V

- Nazwa funkcji jako zmienna.

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

Łańcuchy znaków: Complex Syntax

```
php > $a['cos']=5;
php > echo $a['cos'];
5
php > echo "$a['cos']";
PHP Parse error:  syntax error, unexpected ' ' (T_ENCAPSED_A
php > echo "$a[cos]";
5
php > echo "{$a[cos]}";
PHP Notice:  Use of undefined constant cos - assumed 'cos'
5
php > echo "{$a['cos']}";
5
php > echo "${a['cos']}";
5
```

Przykładowa Obsługa Formularza I

Dane klienta w polach formularza: DATA_*

Zakupione produkty: PROD_*

```
<html>
...
<body>
<?php
$to = 'kowalski@serwer.poczta.pl';
$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";
        }
    }
}
```

Przykładowa Obsługa Formularza II

```
    }  
  }  
}  
$body="{body}\n";  
foreach ( $_POST as $k => $v ) {  
  if (!strncmp($k, 'PROD_', 5)) {  
    if ($v != '') {  
      $n=substr($k, 5);  
      $body="{body}produkt: $n ilosc: $v \n";  
    }  
  }  
}  
$body="{body}\n  
    Razem do zapłaty:$_POST[TOTAL]";  
if (mail($to, $subject, $body))
```

Przykładowa Obsługa Formularza III

```
{
?>
    <p>
    Dziękuję za złożenie zamówienia<br>
<?php
}
else {
?>
    <p>
    Błąd!!! Zamówienie nie zostało wysłane !!!
<?php
}
?>
</body>
</html>
```

UTF8

- Uwaga: łańcuchy znaków w PHP są ciągami bajtów.
- Istnieje zbiór funkcji do przetwarzania ciągów dla kodowań wielobajtowych: `mb_`.

```
echo strlen('abół');
```

6

```
echo mb_strlen('abół');
```

4



Podręcznik PHP.

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

I. Wojnicki, PHP