

Programowanie Aplikacji Internetowych



PHP

Podstawowe informacje :

- język skryptowy
- nie są kompilowane
- wykonywane przez interpreter PHP
- po stronie serwera www, CLI lub GUI
- AMP, LAMP, WAMP
- OpenSource

PHP

Podstawowe informacje :

- żądanie z przeglądarki trafia do serwera www
- dokument trafia do interpretera
- wyszukuje wstawki PHP
- zastępuje je wynikiem ich przetworzenia
- utworzony kod HTML jest zwracany serwerowi
- serwer wysyła go do klienta

PHP

Podstawowe informacje :

- kod PHP nigdy nie opuszcza serwera

```
<html>
<body>
<?php
    echo 'Podaj login';
?>
</body>
</html>
```

```
<html>
<body>
Podaj login
</body>
</html>
```

PHP

Podstawowe informacje :

- kod PHP nigdy nie opuszcza serwera

```
<html>
<body>
<?php
    echo 'Podaj login';
?>
</body>
</html>
```

```
<html>
<body>
Podaj login
</body>
</html>
```

PHP

Polecenie echo:

```
1. <?php
2. echo
3.     'Hello world!';
4. ?>
```

PHP

Polecenie echo:

```
1.  <?php
2.  echo 'To jest tekst 1';
3.  echo 'To jest tekst 2';
4.  echo 'A to jest tekst 3';
5.  ?>
```

PHP

Komentarze:

- systematyzuje całość
- bardzo pomocne przy rozbudowanych algorytmach
- jak działa? jakie ma wymagania?
- komentarze są ignorowane przez interpreter
- służą tylko programiście

PHP

Komentarze:

```
<?php
/*
    komentarz wieloliniowy
    może być rozbijany na wiele linii.
    Cały ten tekst jest ignorowany przez interpreter
*/

// to jest komentarz jednoliniowy - obowiązuje do końca danej linii
# to jest jeszcze jeden komentarz jednoliniowy

?>
```

PHP

Zmienne:

- wszystkie programy komputerowe operują na danych
- w PHP też istnieją mechanizmy ich przechowywania
- mamy różne typy danych

Typy danych:

- skalarne
- złożone (tablice i obiekty)
- specjalne (NULL, resource)

PHP

Skalarne - liczby całkowite

```
1. <?php
2. $a = 1234; // liczba całkowita
3. $a = -123; // liczba całkowita ujemna
4. $a = 0123; // zapis ósemkowy (odpowiednik dziesiętnego 83)
5. $a = 0x1A; // zapis szesnastkowy (odpowiednik dziesiętnego 26)
```

PHP

Skalarne - liczby zmiennoprzecinkowe

```
1.  <?php
2.  $a = 1.234;
3.  $a = 1.2e3;
4.  $a = 7E-10;
```

PHP

Skalarne – typ logiczny oraz teksty

```
1. <?php
2. echo 'To jest tekst zapisany w apostrofach. Kody formatujące pozwalają
3.    umieścić w tekście wyłącznie inne apostrofy: \' albo backslashe: \\. Wszystko inne,
4.    np. \n zostanie wyświetlone jako zwyczajny tekst, zamiast znaku nowej linii';
```

PHP

Operatory

Operator	Nazwa	Składnia
/	Dzielenie	wyrażenie / wyrażenie
%	Dzielenie modulo	wyrażenie % wyrażenie
*	Mnożenie	wyrażenie * wyrażenie
+	Dodawanie	wyrażenie + wyrażenie
-	Odejmowanie	wyrażenie - wyrażenie
.	Łączenie (Konkatenacja)	wyrażenie . wyrażenie
++	Postinkrementacja (zwiększenie)	\$zmienna++
++	Preinkrementacja (zwiększenie)	++\$zmienna
--	Postdekrementacja (zmniejszenie)	\$zmienna--
--	Predekrementacja (zmniejszenie)	--\$zmienna

PHP

Operatory

Operator	Nazwa	Składnia
/	Dzielenie	wyrażenie / wyrażenie
%	Dzielenie modulo	wyrażenie % wyrażenie
*	Mnożenie	wyrażenie * wyrażenie
+	Dodawanie	wyrażenie + wyrażenie
-	Odejmowanie	wyrażenie - wyrażenie
.	Łączenie (Konkatenacja)	wyrażenie . wyrażenie
++	Postinkrementacja (zwiększenie)	\$zmienna++
++	Preinkrementacja (zwiększenie)	++\$zmienna
--	Postdekrementacja (zmniejszenie)	\$zmienna--
--	Predekrementacja (zmniejszenie)	--\$zmienna

PHP

Operatory

```
1. <?php
2. // najpierw składnia $zmienna++
3.
4. $zmienna = 5;
5.
6. echo 'Stan 1: ' . ($zmienna++) . '<br/>';
7. echo 'Stan 2: ' . $zmienna . '<br/><br/>';
8.
9. // teraz składnia ++$zmienna
10. echo 'Restart zmiennej...<br/>';
11. $zmienna = 5;
12.
13. echo 'Stan 1: ' . (++$zmienna) . '<br/>';
14. echo 'Stan 2: ' . $zmienna . '<br/>';
```


PHP

Operatory jednoargumentowe

Operator	Składnia	Równoważna postać
<code>/=</code>	<code>\$zmienna /= wyrażenie</code>	<code>\$zmienna = \$zmienna / wyrażenie</code>
<code>%=</code>	<code>\$zmienna %= wyrażenie</code>	<code>\$zmienna = \$zmienna % wyrażenie</code>
<code>*=</code>	<code>\$zmienna *= wyrażenie</code>	<code>\$zmienna = \$zmienna * wyrażenie</code>
<code>+=</code>	<code>\$zmienna += wyrażenie</code>	<code>\$zmienna = \$zmienna + wyrażenie</code>
<code>-=</code>	<code>\$zmienna -= wyrażenie</code>	<code>\$zmienna = \$zmienna - wyrażenie</code>
<code>.=</code>	<code>\$zmienna .= wyrażenie</code>	<code>\$zmienna = \$zmienna . wyrażenie</code>

PHP

Operatory jedno argumentowe

```
1. <?php
2.
3. $tekst = 'Litwo, ojczyzno moja! Ty jesteś jak zdrowie<br/>';
4. $tekst .= 'Ile Cię trzeba cenić, ten tylko się dowie<br/>';
5. $tekst .= 'Kto Cię stracił, dziś piękność twą w całej ozdobie<br/>';
6. $tekst .= 'Widzę i opisuję, bo tęsknię po tobie.<br/>';
7.
8. echo $tekst;
```

PHP

Tablica to zmienna zbiorcza, grupująca pojedyncze elementy mające właściwości zmiennych, do których odwoływać możemy się za pomocą indeksów.

```
1. <?php
2. $tablica = array(0 => 5, 3, 8, 7, 9, 24, 15, 2, 19);
3. $liczba_losowa = rand(0, 8);
4. echo 'Pod numerem '.$liczba_losowa.' kryje się wartość '.$tablica[$liczba_losowa];
```

PHP

Tablica to zmienna zbiorcza, grupująca pojedyncze elementy mające właściwości zmiennych, do których odwoływać możemy się za pomocą indeksów.

```
1. <?php
2. $tablica = array(0 => 5, 3, 8, 7, 9, 24, 15, 2, 19);
3. $liczba_losowa = rand(0, 8);
4. echo 'Pod numerem '.$liczba_losowa.' kryje się wartość '.$tablica[$liczba_losowa];
```

PHP - wprowadzenie

warto przećwiczyć

Ćw. 1 Przy użyciu poznanych elementów języka stwórz dwie zmienne. Pierwszą przechowującą imię i nazwisko oraz drugą, która przechowuje wiek. Całość wyświetl na ekran w jednej linii.

Ćw. 2 Wykorzystując poznaną funkcję `rand()` zmodyfikuj program, aby wiek zmieniał się w sposób losowy.

W razie problemów skorzystaj z php.net/rand.



PHP

Programy wykonujące szereg zawsze tych samych instrukcji nie są interesujące. Prawdziwa aplikacja musi umieć reagować na poczynania użytkownika. Tu do akcji wkraczają instrukcje kontrolne. Są one podstawowym budulcem wielu algorytmów i ich poznanie jest niezbędne.

PHP

```
1. <?php
2. if(wyrazenie)
3. {
4.     // blok kodu
5. }
6. elseif(wyrazenie)
7. {
8.     // blok kodu
9. }
10. else
11. {
12.     // blok kodu
13. }
```

PHP

```
1. <?php
2. if($zmienna == 6)
3.     echo 'Wartość zmiennej wynosi 6';
```


PHP

Operator	Nazwa	Składnia	Opis
==	Równość	wyrażenie == wyrażenie	Zwraca prawdę, jeżeli oba wyrażenia mają identyczną wartość.
===	Równość	wyrażenie === wyrażenie	Zwraca prawdę, jeżeli oba wyrażenia mają identyczną wartość oraz typ .
!=	Nierówność	wyrażenie != wyrażenie	Zwraca prawdę, jeżeli oba wyrażenia mają różne wartości.
!==	Nierówność	wyrażenie !== wyrażenie	Zwraca prawdę, jeżeli oba wyrażenia mają różne wartości i/lub typ.
<	Mniejsze niż	wyrażenie < wyrażenie	Zwraca prawdę, jeżeli lewe wyrażenie ma mniejszą wartość od prawego.
>	Większe niż	wyrażenie > wyrażenie	Zwraca prawdę, jeżeli lewe wyrażenie ma większą wartość od prawego.
<=	Mniejsze lub równe	wyrażenie <= wyrażenie	Zwraca prawdę, jeżeli lewe wyrażenie ma mniejszą lub równą wartość prawemu.
>=	Większe lub równe	wyrażenie >= wyrażenie	Zwraca prawdę, jeżeli lewe wyrażenie ma większą lub równą wartość prawemu.
!	Negacja (nie)	!wyrażenie	Zwraca prawdę, jeżeli wyrażenie jest fałszywe i fałsz, jeśli prawdziwe.
&&	Koniunkcja logiczna (i)	wyrażenie && wyrażenie	Zwraca prawdę, jeżeli oba wyrażenia są prawdziwe.
	Alternatywa logiczna (lub)	wyrażenie wyrażenie	Zwraca prawdę, jeżeli przynajmniej jedno z wyrażeń jest prawdziwe.

PHP

Oprócz operatorów warunkowych przydatne bywają funkcje:

- `isset($val)`
- `empty($val)`
- `is_null($val)`
- `is_string`, `is_float`, `is_numeric`, `is_bool`, `is_array`

PHP - wprowadzenie

warto przećwiczyć

Ćw. 3 Zmodyfikuj aplikację nr 2, tak aby w przypadku wyświetlenia wieku mniejszego niż 18 wyświetlał stosowny komunikat.

Ćw. 4 Wykorzystując poznaną funkcję `rand()` zmodyfikuj program, aby wiek zmieniał się w sposób losowy.

W razie problemów skorzystaj z php.net/rand.



PHP

Oprócz operatorów warunkowych przydatne bywają funkcje:

- `isset($val)`
- `empty($val)`
- `is_null($val)`
- `is_string`, `is_float`, `is_numeric`, `is_bool`, `is_array`

SWITCH

```
switch($_GET['act'])
{
    case 'dodaj':
        echo 'Dodawanie danych';
        break;
    case 'edytuj':
        echo 'Edycja danych';
        break;
    case 'usun':
        echo 'Usuwanie danych';
        break;
    default:
        echo 'Wyświetlenie danych';
}
```

PHP

Pętle:

- for
- while
- do while
- foreach

FOR

```
1.  <?php
2.  for($i = 0; $i < 10; $i++)
3.  {
4.      echo $i.'<br/>';
5.  }
```

FOR

```
1.  <?php
2.  for($i = 0; $i < 10; $i++)
3.  {
4.      echo $i.'<br/>';
5.  }
```


WHILE

```
1. <?php
2. while(rand(0,10) != 8)
3. {
4.     echo 'Jeszcze nie wylosowałem!<br/>';
5. }
```

WHILE

```
1.  <?php
2.  $i = 0;
3.  while($i < 10)
4.  {
5.      echo $i.'<br/>';
6.      $i++;
7.  }
```

FOREACH

```
foreach($plik as $numer => $linia)
{
    echo '<li>Linia #' . $numer . ': ' . trim($linia) . '</li>';
}
```

DO WHILE

```
1. <?php
2. do
3. {
4.     echo "Podaj i: \n";
5.     fscanf(STDIN, "%d\n", $i); // 1
6. }
7. while($i < 10);
```