



Podstawy php

G. Pręczonek

- **zmienna** - to obiekt w programowaniu, który przechowuje różnego rodzaju dane niezbędne do działania programu. Zmienna podczas działania programu może zmieniać swoje wartości (jak wskazuje nazwa). Tworząc zmienną musimy nadać jej nazwę oraz typ, który określa co nasza zmienna będzie przechowywać.

Nadając nazwę trzymamy się następujących reguł:

- zmienna jest jednym ciągiem znaków bez spacji np. nazwa_zmiennej - dobrze, nazwa zmiennej - źle
- nie zaczynamy nazwy od cyfry np. 12zmienna - źle, zmienna12 - dobrze
- nie używamy polskich liter takich jak ą, ę itp.
- nazwa zmiennej powinna kojarzyć się z przeznaczeniem tej zmiennej np. tablica_ciagu - dobrze
- nazwa nie może być słowem kluczowym języka programowania np. auto - źle

Zmienne

Zmienne w Php nie są deklarowane

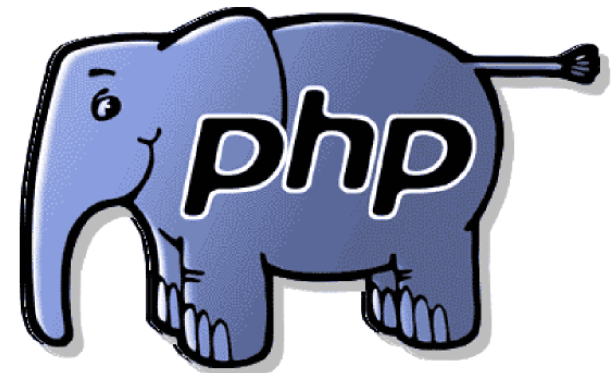
Przed nazwą zmiennej występuje znak \$

Zmienne nie muszą być definiowane na początku kodu
mogą być definiowane w każdym miejscu kodu

Zmienną możemy wprowadzać w dowolnej części skryptu. Operator = przypisuje zmiennej określoną wartość

Wartość zmiennej możemy wydrukować za pomocą instrukcji echo

```
$a = 1;  
echo $a;
```



Zasięg zmiennych

- **Zmienne poziomu funkcji** (definiowane wewnątrz funkcji i poprawne tylko dla niej)
- Zmienne deklarowane w skrypcie głównym -- **globalne**
- Zmienne superglobalne

Typy zmiennych

- **Typ logiczny bool**

Po wstawieniu do zmiennej wartości logicznej, typem zmiennej będzie bool:

```
<?php
```

```
$widoczny = false;
```

```
?>
```

Typ całkowity int

Po wstawieniu do zmiennej wartości całkowitej, typem zmiennej będzie int:

```
$liczba = 619;
```

Jeśli do zmiennej wstawimy wartość **zmiennopozycyjną**:

```
$wspolczynnik = 3.1415;
```

to typem zmiennej będzie **float**:

Napisy. Typ string

Jeśli natomiast do zmiennej wstawimy napis:

```
$marka = 'Opel';
```

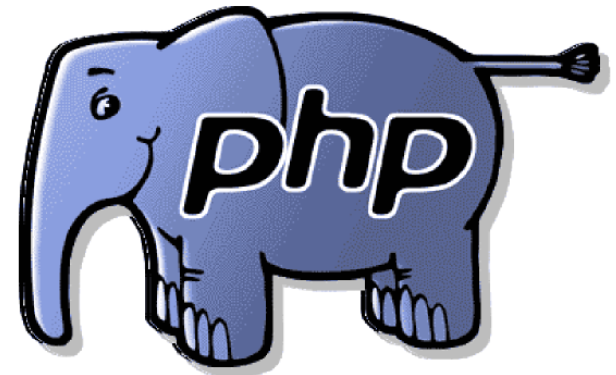
Wartość i typ zmiennej możemy sprawdzić za pomocą funkcji `var_dump()`

W wyniku wykonania skryptu

```
<?php  
$dlugosc = 1234;  
var_dump($dlugosc);  
?>
```

Otrzymamy
int 1234

Oznacza on, że zmienna o nazwie `$dlugosc` przechowuje liczbę całkowitą (napis int) o wartości 1234



Wyrażeniem jest każda instrukcja, która zwraca wartość

Np. instrukcja przypisania

`$a = 5`

Zwraca wartość 5 W ten sposób instrukcja przypisania może wystąpić jako fragment większego wyrażenia

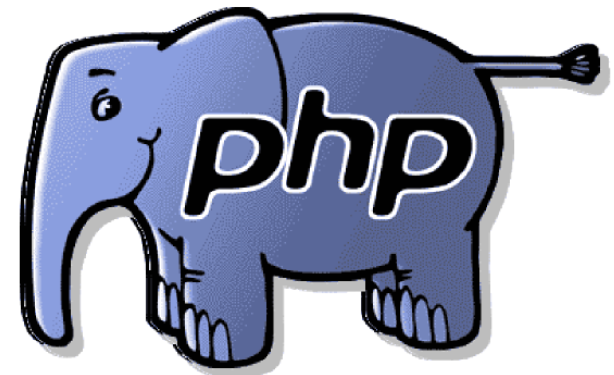
`$b = ($a = 5) + 7;` zmienna wynikowa

Powyższa instrukcja spowoduje najpierw przypisanie wartości 5 do zmiennej \$a. wartość 5 zwrócona przez pierwsze przypisanie zostanie powiększona o 7 i wstawiona do zmiennej b. ostatecznie zmienna b przyjmuje wartość 12. Dokładnie taki sam efekt uzyskamy stosując 2 instrukcje

`$a = 5;`

`$b = $a + 7;`

`echo ($b);`



Proste operacje na zmiennych - operatory

OPERATORY:

//operator przypisania – wykorzystywany do przypisania określonych wartości

```
$licznik = 0;
```

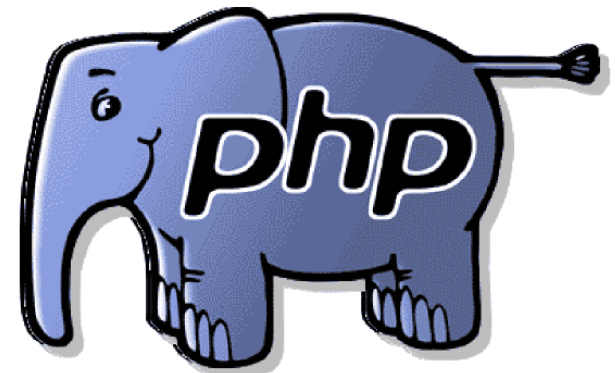
```
$Licznik = "5";
```

```
$ilosc = 7;
```

```
$informacja = "Stan licznika wynosi ";
```

```
$wlaczony = true;
```

```
$licznik = $ilosc;
```



// operatory arytmetyczne

echo \$ilosc + 15; **dodawanie +**

\$ilosc = \$ilosc + 15;

\$wynik1 = \$ilosc - \$licznik;

\$wynik2 = \$ilosc * \$Licznik;

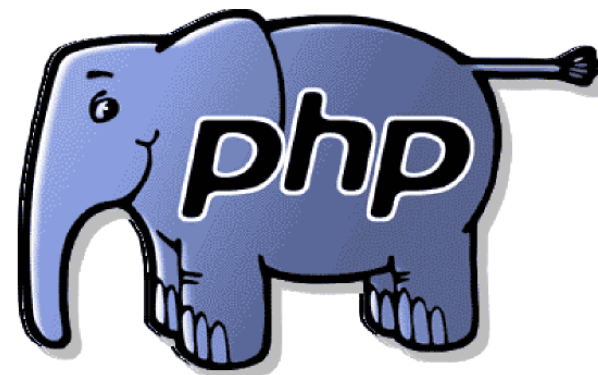
\$wynik3 = 150 / \$ilosc;

\$ilosc = 2;

\$reszta = 15 % \$ilosc;

echo 'Reszta z dzielenia 15 / ' . \$ilosc . ' wynosi ' . \$reszta;

Operator



Operatory porównania

Przykład	Nazwa	Wynik
$\$a == \b	Równy	Prawda jeśli $\$a$ jest równe $\$b$.
$\$a === \b	Identyczny	Prawda jeśli $\$a$ jest równe $\$b$ i są tego samego typu. (tylko PHP4)
$\$a != \b	Nie równe	Prawda jeśli $\$a$ nie jest równe $\$b$.
$\$a !== \b	Nie identyczny	Prawda jeśli $\$a$ nie jest równe $\$b$ lub nie są tego samego typu. (tylko PHP4)
$\$a < \b	Mniejsze	Prawda jeśli $\$a$ jest mniejsze niż $\$b$.
$\$a > \b	Większe	Prawda jeśli $\$a$ jest większe niż $\$b$.
$\$a <= \b	Mniejsze lub równe	Prawda jeśli $\$a$ jest mniejsze lub równe $\$b$.
$\$a >= \b	Większe lub równe	Prawda jeśli $\$a$ jest większe lub równe $\$b$.

//Operator kontroli błędów

- Operator kontroli błędów ('@') powoduje, że wyrażenie przed którym postawiono ten znak nie spowoduje wyświetlenia się jakiegokolwiek błędu lub ostrzeżenia.

Operatory logiczne

Operatory logiczne służą do budowania bardziej skomplikowanych instrukcji warunkowych – do łączenia kilku warunków w jednej instrukcji.

Przykład	Nazwa	Wynik
<code>\$a && \$b</code>	AND	Prawda, jeśli <code>\$a</code> i <code>\$b</code> są prawdą
<code>\$a \$b</code>	OR	Prawda, jeśli <code>\$a</code> lub <code>\$b</code> są prawdą
<code>! \$a</code>	NOT	Prawda, jeśli <code>\$a</code> nie jest prawdą

// inkrementacija i dekrementacija

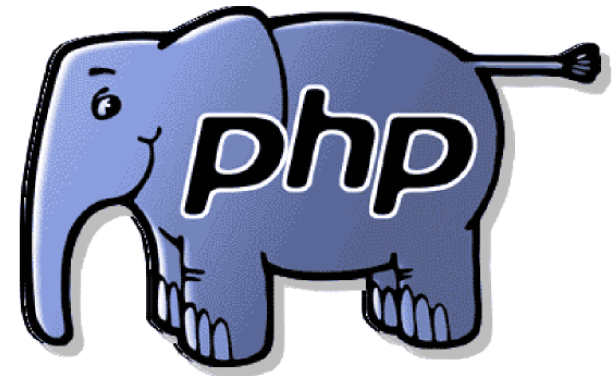
```
$a = 1;
```

```
//++$a; //preinkrementacija
```

```
//$a++; //postinkrementacija
```

```
echo $a++; echo "<br>";
```

```
echo $a;
```



STAŁE

W języku php stałe definiujemy wywołując funkcję **Define**

Ma ona 3 parametry:

\$name

\$value

\$case_insensitive

Dwa pierwsze są wymagane trzeci opcjonalny

Parametr \$name jest napisem. Ustala on nazwę definiowanej stałej

Parametr \$value może być dowolnego typu skalarnego

