

APLIKACJE INTERNETOWE PHP



TEMAT 4-03: Operacje na obiektach.

Autor dokumentu: Wojciech Galiński

piątek, 9 grudnia 2016 r.

351203 Technik informatyk

ŹRÓDŁA WIEDZY:

<http://www.php.net/manual/pl>, <http://webmaster.helion.pl/index.php/kurs-php>, <http://pl.wikipedia.org/>,
<http://pl.wikibooks.org/wiki/PHP>, <http://phpkurs.pl/>, <http://kursphp.com/>.

Zagadnienia obowiązkowe

1. **Sprawdzanie, czy zmienna jest obiektem** – warunek ten sprawdzamy za pomocą funkcji: „**is_object**”.

PRZYKŁAD:

```
class Pusta {};  
$obiekt_1 = new Pusta;   $obiekt_2 = 5;  
var_dump(is_object($obiekt_1), is_object($obiekt_2));
```

2. **Serializacja** – to konwersja tabeli lub obiektu do tekstu. Można również wykonywać operację odwrotną do serializacji, co przedstawiono w poniższych przykładach.

Serializację można wykorzystać do przechowywania zawartości obiektów np. w bazie danych.

PRZYKŁAD SERIALIZACJI TABLICY:

```
$tablica = array(10, 11, 12, 'a'=>'x', 'b'=>'y', 'c'=>'z');  
var_dump($tablica);   echo '<br />'; // tak wygląda ta tablica  
$tekst = serialize($tablica);  
echo $tekst. '<br />'; // a tak po serializacji  
echo '<pre>';  
var_dump(unserialize($tekst)); // wynik cofnięcia serializacji  
echo '</pre>';
```

PRZYKŁAD SERIALIZACJI OBIEKTU:

```
class Klasa_do_serialiacji  
{  
    const TEKST = 'tekst';   public static $ilosc = 5.5;  
    public $liczba = 3;  
    public function funkcja() { echo $this->liczba. ' '. self::TEKST; }  
}  
$obiekt = new Klasa_do_serialiacji();  
echo '<pre>';   var_dump($obiekt);   echo '</pre>';  
$tekst = serialize($obiekt);   echo $tekst;  
echo '<pre>';   var_dump(unserialize($tekst));   echo '<pre>';
```

3. **Referencje w PHP** – to nazwy bytów przechowywanych w pamięci komputera podczas działania skryptu PHP. Pojedynczy byt może mieć wiele referencji.

Obiekty w PHP są referencjami, dzięki czemu kilka różnych nazw może odnosić się do tego samego bytu. Podczas przypisywania jednego obiektu do drugiego, kopiowana jest referencja, a nie zawartość całego obiektu. Trwa to szybko, ale operujemy cały czas na jednej kopii danych.

PRZYKŁAD 1:

```
$a = 5;  
$b = &$a;  
$b = 3;  
echo '<p>'. $a. ' '. $b. '</p>';
```

PRZYKŁAD 2:

```
class Jakas_klasa { public $wartosc = 1; };  
$ob_1 = new Jakas_klasa();   $ob_1->wartosc = 5;   $ob_2 = $ob_1;  
$ob_2->wartosc = 10;   echo $ob_1->wartosc. ' '. $ob_2->wartosc. '<br />';
```

4. **Klonowanie obiektu (instrukcja „clone”)** – to tworzenie kopii obiektu (patrz też: punkt 3).

Aby utworzyć nowy obiekt i skopiować do niego całą zawartość innego obiektu należy użyć słowa kluczowego „**clone**”.

PRZYKŁAD: (kontynuacja przykładu 2 z poprzedniego punktu):

```
$ob_3 = new Jakas_klasa();   $ob_3->wartosc = 5;   $ob_4 = clone $ob_3;  
$ob_4->wartosc = 10;   echo $ob_3->wartosc. ' '. $ob_4->wartosc;
```

5. **Słowo kluczowe „friend” w PHP** – obecnie język PHP 7.0 nie obsługuje przyjaźni z klasami.

Zadania

1. Zapoznaj się z przedstawionymi przykładami. Wykonaj je i przeanalizuj ich działanie.
2. Utwórz klasę realizującą wyświetlanie planu lekcji jako tabelę HTML, zamień tę klasę na tekst i na odwrót, a następnie sklonuj obiekt tej klasy.