

APLIKACJE INTERNETOWE PHP



TEMAT 1-04: Pętle w PHP.

Autor dokumentu: Wojciech Galiński

piątek, 16 września 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. **Instrukcja iteracyjna (pętla)** – to instrukcja, która umożliwia wielokrotne wykonywanie się pojedynczego zestawu instrukcji. Pomaga ona w realizowaniu powtarzających się zadań np. 10 razy wyświetlić coś na ekranie.
2. **Iteracja (powtórzenie)** – to jednokrotne wykonanie zestawu instrukcji wewnątrz instrukcji iteracyjnej (pętli).
3. **Pętla „while”** – oto jej ogólna postać:

```
while (warunek) instrukcja;
```

```
while (warunek)
{
    // instrukcje
}
```

Jej konstrukcja jest bardzo podobna do instrukcji warunkowej „if”. Instrukcja lub instrukcje wykonywane są, dopóki warunek jest spełniony. Warunek jest sprawdzany **przed** każdą iteracją, więc instrukcje wewnątrz tej pętli mogą nie wykonać się ani razu (tak jak w instrukcji „if”).

PRZYKŁADY:

```
$i=0; while ($i++<100) echo $i. ' '; $j=0; while (++$j>0) echo $i. ' ';
```

4. **Pętla „for”** – to rozbudowana wersja pętli „while”. Oto jej ogólna postać:

```
for (wyrazenie_i; warunek; wyrazenie_m) instrukcja;
```

albo

```
for (wyrazenie_i; warunek; wyrazenie_m)
{
    // instrukcje
}
```

Oto sposób działania tej pętli:

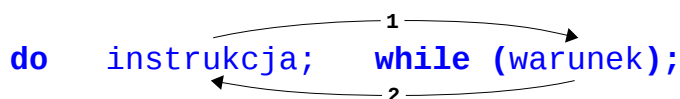
- ➔ najpierw wykonywane jest wyrażenie inicjalizujące („wyrazenie_i”) – dotyczy ono zmiennych, które zliczają kolejne iteracje, a zwane są iteratorami (zwykle występuje tam 1 zmienna);
- ➔ następnie wykonywane jest wyrażenie warunkowe („warunek”);
- ➔ gdy warunek nie jest spełniony, następuje wyjście z pętli „for” do kolejnej instrukcji;
- ➔ gdy warunek jest spełniony, wykonane są: instrukcje wewnątrz pętli oraz wyrażenie modyfikujące iterator („wyrazenie_m”);
- ➔ kolejna iteracja rozpoczyna się od sprawdzenia warunku (wyrażenie inicjalizujące wykonywane jest tylko jednokrotnie – na samym początku pętli).

PRZYKŁADY:

```
for ($i=0; $i<10; ++$i) echo $i. ' ' ;
for ($i=1; $i<=9; ++$i) echo $i. ' ' ;
for (; $i>=1; --$i) echo $i. ' ' ;
```

```
$a=20; $b=100;
for ($i=$a, $j=$b; $i<$j; $i+=5, $j-=5) echo $i. ' ' ;
```

5. **Pętla „do ... while”** – oto jej ogólna postać:



```
do
{
    // instrukcje
}
while (warunek);
```

Instrukcja lub instrukcje wykonywane są, dopóki warunek jest spełniony, ale warunek sprawdzany jest **po** każdej iteracji, więc instrukcje wewnątrz tej pętli muszą się wykonać co najmniej jeden raz.

PRZYKŁADY:

```
$i=0; do echo $i. ' '; while ($i++<100);
$j=0; do echo $j. ' '; while (++$j>0); // UWAGA: pętla nieskończona – skrypt się zawiesi!
```

6. **Pętla „foreach”** – patrz: temat „Tablice”.

7. **Zagnieżdżanie pętli** – to umieszczanie jednej pętli wewnątrz innej (mogą być to 2 lub więcej dowolnych pętli – niekoniecznie „for”).

PRZYKŁAD:

```
for ($x=-1; $x<2; ++$x)
    for ($y=-1; $y<2; ++$y)
        for ($z=-1; $z<2; ++$z) echo '['. $x. ', ' . $y. ', ' . $z. ']' );
```

Zagnieżdżanie może spowodować długie wykonywanie się skryptu, np.

PRZYKŁAD:

```
for ($i=0; $i<1000; ++$i)
    for ($j=0; $j<1000; ++$j) echo '['. $i. ', ' . $j. ']' );
```

spowoduje aż milion (!) iteracji (to tak jakby odwołać się do wszystkich elementów arkusza kalkulacyjnego o rozmiarach 1000x1000 komórek).

8. **Przerywanie bieżącej iteracji (instrukcja „continue”)** – służy do tego instrukcja „continue”. Po napotkaniu takiej instrukcji natychmiast rozpocznie się wykonywać nowa iteracja pętli.

PRZYKŁAD:

```
for ($i=0; $i<10; ++$i)
    if ($i%3) continue;
    else echo $i. ' ';
```

9. **Przerywanie pętli (instrukcja „break”)** – służy do tego instrukcja „break”. Po napotkaniu takiej instrukcji następuje natychmiastowe przeniesienie wykonywania programu do pierwszej instrukcji poza pętlą (ale pamiętać trzeba, że w pętli zagnieżdżonej powoduje ona wyjście tylko o 1 zagnieżdżenie do góry, na co trzeba zwracać uwagę w pętlach zagnieżdżonych), np.

PRZYKŁAD:

```
for (;;) break;
for ($i=0; $i<10; ++$i)
    for ($j=0; $j<10; ++$j)
        if ($j<=2) echo $i. ' ' ;
        else break;
```

Zadania

- Przekaż w adresie skryptu następujące parametry: **N** – liczba zmiennych, **x1**, ..., **xN** – zmienne liczbowe. Wyświetl operację sumowania tych liczb, np. dla parametrów: „skrypt.php?N=3&x1=4&x2=-6&x3=10” wyświetl tekst: „4 + (-6) + 10 = 8”.
- Prześlij do skryptu 3 parametry liczbowe: **a**, **b**, **krok**. Następnie wyświetl wszystkie liczby z przedziału [a,b] co „krok” (rozdzielając je przecinkiem). Jeśli a>b, liczby wyświetlaj malejąco. Pomiń wyświetlanie liczb podzielnych przez 3 i 7. PRZYKŁAD: „[35, 1; 5]: 25, 20, 10, 5”.
- Wyświetl w tabeli HTML tabliczkę potęgowania od 1 do 5 (użyj do tego jednej z poznanych pętli). Pierwszy rząd i pierwsza kolumna mają być nagłówkami tabeli.
- Wygeneruj formularz z 50 kontrolkami typu „input radio” i opisem do każdej „Opcja X” (X – numer kontrolki). Zadbaj, żeby formularz działał prawidłowo i żeby wystarczyło kliknąć w opis kontrolki w celu jej zaznaczenia. Oto przykładowy wygląd kontrolki:
☐ Opcja 1 .