**Instrukcje sterujące w PHP cz.II**

**Film nr 18, 19**

***Pętla for***

Konstrukcja pętli for jest następująca:

for (wyrażenie1; warunek; wyrażenie2)

{

blok\_instrukcji;

}

blok\_instrukcji będzie wykonywany dopóty, dopóki spełniony jest warunek. Mówiąc inaczej - najpierw interpreter sprawdza, czy warunek jest spełniony. Jeżeli tak, to wykonywany jest blok\_instrukcji. Jeżeli nie, to wykonywanie pętli jest zakończone i wykonywane są instrukcje znajdujące się po bloku pętli.

*Przykład* Napisz skrypt, który wyświetli 1000 kolejnych liczb całkowitych począwszy od 5. Zastosuj jeden plik.

<?php

for ($i=1; $i<=1000; $i++)

{

$liczba=$i+4;//ponieważ zaczynamy wyświetlanie od 5

if ($i<1000) $napis=$liczba.', ';//oddzielamy kolejne liczby //przecinkami, oprócz ostatniej

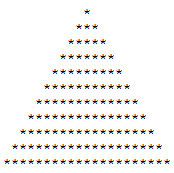
echo $napis;

}

?>

**Zadanie 1** Napisz skrypt, obliczający n! (n silnia). Liczbę n wyślij formularzem. Dla uproszczenia, zakładamy, że użytkownik wyśle liczbę całkowitą, nieujemną.

**Zadanie 2** Napisz skrypt, który za pomocą znaków \* (gwiazdki) narysuje choinkę. Wysokość choinki zdefiniuj za pomocą stałej. Zastosuj jeden plik.



***Wskazówka!*** Ponieważ przeglądarka nie wyświetla więcej niż jednej spacji użyj znacznika <pre>…</pre>. Np.

echo '<pre>'.$choinka.'</pre>';

gdzie $choinka jest zmienną zawierającą spacje i gwiazdki oraz przejścia do nowego wiersza.

***Zagnieżdżanie pętli***

*Przykład* Przetestuj pokazany kod PHP. Jaka będzie wartość zmiennej $a, jeżeli $i==2, $j==3 oraz $k==4?

<?php

$a=0;

for ($i=1; $i<=10; $i++)

for ($j=1; $j<=20; $j++)

for ($k=1; $k<=30; $k++)

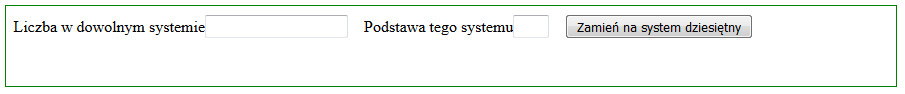
{

$a++;

echo 'Dla $i=='.$i.', $j=='.$j.', $k=='.$k.' wartość $a wynosi '.$a.'<br />';

}

?>

**Zadanie 3** Zamiana liczby z dowolnego systemu o podstawie z przedziału <2;16> na system dziesiętny.

1. przygotuj formularz jak poniżej.
2. zdefiniuj funkcję i przetestuj ją

function wartosc\_znaku($znak)

{

$znaki='0123456789ABCDEF';

return strpos($znaki,$znak);

}

1. pobierz dane z formularza i sprawdź ich poprawność

$liczba=$\_POST['liczba'];

$podstawa=$\_POST['podstawa'];

if ($liczba=='' || $podstawa=='') echo 'Brak danych!';

elseif ($podstawa<2 || $podstawa>16) echo 'Podstawa musi być liczbą z przedziału 2..16 ';

else

{

//tu umieść algorytm przeliczający

}

1. algorytm przeliczający może wyglądać tak:

echo $liczba.'<sub>'.$podstawa.'</sub> = ';

$nowa\_liczba=0;

$licznik=0;

for ($z=strlen($liczba)-1;$z>=0;$z--)

{

$licznik++;

if ($licznik==1) $mnoznik=1; else $mnoznik\*=$podstawa;

$nowa\_liczba+=$mnoznik\*wartosc\_znaku($liczba[$z]);

}

echo $nowa\_liczba.'<sub>10</sub>';

**Film nr 20**

***Pętla while***

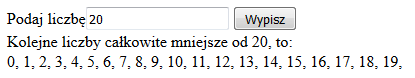
Poznaną już pętlę for, najczęściej stosujemy z licznikiem - pierwsze wyrażenie inicjuje stan początkowy licznika, warunek sprawdza stan aktualny licznika, natomiast drugie wyrażenie definiuje zmianę stanu licznika. Jak więc widzimy, taka postać określa z góry liczbę obiegów pętli. Często jednak zdarzają się sytuacje, kiedy musimy zastosować pętlę, nie wiedząc ile razy się ona wykona. Wystarczy wtedy określić warunek wykonania kolejnego obiegu pętli. Najprostszą pętlą sprawdzającą taki warunek jest

while (warunek)

instrukcja

gdzie instrukcja, może być pojedynczą instrukcją (w tym także pętlą) lub blokiem instrukcji.

**Zadanie 4** Wyświetl po przecinkach, wszystkie liczby całkowite nieujemne, mniejsze od liczby wysłanej formularzem.



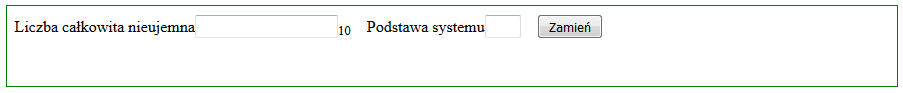
**Bez tytułu.pngZadanie 5** Liczbę całkowitą wysłaną formularzem, wyświetl w odwrotnej kolejności cyfr.

*Wskazówka:*

Dopóki liczba>0 powtarzaj:

* oblicz resztę z dzielenia liczby przez 10, otrzymasz ostatnią cyfrę
* podziel liczbę przez 10 i „obetnij” część ułamkową, skorzystaj z funkcji floor()

**Zadanie 6** Liczbę całkowitą nieujemną wysłaną formularzem, wyświetl w notacji wybranego systemu o podstawie z zakresu <2;16>.

1. Przygotuj formularz jak poniżej.
2. Zdefiniuj funkcję i ją przetestuj

function cyfra($wartosc)

{ $cyfry='0123456789ABCDEF';

return $cyfry[$wartosc];

}

1. pobierz dane z formularza i sprawdź ich poprawność

$liczba=$\_POST['liczba'];

$podstawa=$\_POST['podstawa'];

if ($liczba==0) echo '0';

elseif ($podstawa<2 || $podstawa>16) echo 'Podstawa musi być liczbą z przedziału 2..16 ';

else

{

//tutaj umieść przeliczanie

}

1. algorytm przeliczania może wyglądać tak (zastosowano funkcję strrev() do wypisywania stringu od końca):

echo $liczba.'<sub>10</sub> = ';

$nowa\_liczba='';

while ($liczba>0)

{

$nowa\_liczba.=cyfra($liczba%$podstawa);

$liczba=floor($liczba/$podstawa);

}

echo strrev($nowa\_liczba).'<sub>'.$podstawa.'</sub>';

***Pętle do...while***

Pętla ta działa podobnie jak pętla [while](http://www.teacher.webd.pl/kurs_php/k_1_12_2.php), przy czym instrukcja (lub blok instrukcji jak pokazano poniżej) jest wykonana przynajmniej jeden raz, ponieważ warunek jest sprawdzany po jej wykonaniu a nie, jak w przypadku while, przed wykonaniem instrukcji.

do

{

instrukcja\_1;

instrukcja\_2;

...

instrukcja\_n

}

while (warunek);

*Przykład* Funkcja wbudowana rand($min,$max) generuje liczbę losową z przedziału $min..$max. Napisz skrypt wyświetlający liczby wybrane losowo, do chwili gdy taką liczbą jest 5.

<?php

do

{

$liczba\_losowa=rand(1,20);

echo $liczba\_losowa;

if ($liczba\_losowa!=5) echo ', ';

}

while ($liczba\_losowa!=5);

?>

**Bez tytułu.pngZadanie 7** Zapisz skrypt, który wyświetla wszystkie liczby całkowite od zera do danej formularzem liczby.

**Zadanie 8** Zapisz skrypt, który wyświetla trzy różne liczby całkowite losowane z przedziału <1;10>.

**Film nr 21**

***Zatrzymywanie pętli i skryptu***

* instrukcja break; powoduje natychmiastowe wyjście z pętli i rozpoczęcie wykonywania pierwszej instrukcji znajdującej się po pętli (działanie podobne jak w instrukcji switch);
* instrukcja continue; powoduje przeskoczenie do następnego obiegu pętli;
* instrukcja exit; powoduje zakończenie wykonywania skryptu.

*Przykład*

<?php

for ($i=1; $i<=1000; $i++)

{

echo $i;

if ($i<10) echo ', '; else **break**;

}

echo '<br />Instrukcja wykonywana po zakończeniu pętli';

?>

<?php

for ($i=1; $i<=20; $i++)

{

if ($i%2==0) echo $i; else **continue**;

if ($i<20) echo ', ';

}

?>

<?php

$a=$\_POST['dzielna'];

$b=$\_POST['dzielnik'];

if ($b==0)

{

echo 'Nie ma dzielenia przez zero!';

**exit;**

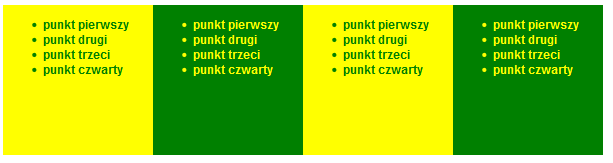
}

echo "$a/$b = ".$a/$b;

?>

***Alternatywne składnie struktur sterujących***

Podczas pisania kodu PHP bardzo często występują sytuacje, kiedy musimy ten kod przeplatać z kodem HTML. Stosowanie wtedy nawiasów { i } oraz odpowiednich wcięć, nie tylko nie poprawi przejrzystości kodu ale go "popsuje". W takich sytuacjach możemy zamiennie zastosować składnię w której nawias otwierający { zastąpimy dwukropkiem :, natomiast nawias zamykający } odpowiednio endif, endswitch, endwhile, endfor lub endforeach. Dla pętli do...while nie ma składni alternatywnej.

**Zadanie 9** Zapisz skrypt, który wyświetli zdefiniowaną stałą ilość kolorowych bloków div (szerokość i wysokość wynosi 150px) zawierających listy ul. Poniżej podano przykład dla 4 bloków div.

**Zadanie 10** Zapisz skrypt, który wyświetli tabelę o dwóch kolumnach i n wierszach (n – zdefiniowane jako stała). Tabela powinna zawierać wiersz nagłówkowy, kolejne liczby 0..n oraz obliczone z nich silnie. Tło wierszy powinno zmieniać się na zmianę co drugi wiersz (białe, szare).