Young Programmer

www.inceptio.org.pl

Dr inż. Małgorzata Janik

Organizator:

fundacja INCEPTIO

Partnerzy:

Wydział Fizyki Politechnika Warszawska

Stowarzyszenie KLATRAT

Dofinansowanie:

MIASTO STOŁECZNE WARSZAWA

Zatrzymaj się w Warszawie

Ramowy program warsztatów

Zajęcia 1: Zajęcia wprowadzające, HTML

Zajęcia 2: Style CSS

Zajęcia 3: Podstawy języka PHP

Zajęcia 4: Formularze HTML

Zajęcia 5: Język PHP cd.

Zajęcia 6: Synteza kursu

http://www.if.pw.edu.pl/~majanik/wiki/index.php/HTML%2BPHP

Podstawy PHP

Dr inż. Małgorzata Janik

Umieszczanie kodu PHP

```
<!-- jakiś kod HTML -->

<?php

// Kod php
echo 'Witaj cudowny świecie!';

?>

<!-- jakiś kod HTML →

<?php

echo 'Znowu PHP!';

?>
```

Kod PHP można umieszczać w plikach *.php naprzemiennie z kodem HTML dowolną liczbę razy.

Podstawowe funkcje – wyświetlanie

```php
<?php

// Wyświetlanie tekstu
echo '<b>Witaj piękny świecie!</b>';
echo "Wartość obliczonej reszty wynosi $reszta.";

// Wyświetlanie liczb
echo 44;
echo 12*$tuzin;

?>
```

Napisy wyświetlone przez funkcję echo są „wklejane” w miejsce kodu PHP w pliku wynikowym. Często więc będziemy stosować w nich tagi HTML.

Zmienne i operacje arytmetyczne

```php
<?php

$moja_zmienna = 33;              // nowa zmienna liczbowa
$kaczor = 'Donald';              // nowa zmienna tekstowa
$roznica = $moja_zmienna - 15;   // odejmowanie
$moja_zmienna++;                 // dodaje 1
$kaczor2 = $kaczor.' Duck';      // łączenie napisów
$tuzin = 12;
$cztery_tuziny = 4 * $tuzin;     // mnożenie
$reszta = 46 % 3;                // reszta z dzielenia 46 przez 3

?>
```

Operacja arytmetyczne można wykonywać „osobno” (jak w powyższych przykładach albo we wszystkich innych miejscach, w których można wstawić samą wartość.

Tablice

```php
<?php

// Tablica o 4 elementach
$uczen[1] = 'Jan Kowalski';
$uczen[2] = 'Jan Nowak';
$uczen[3] = 'Piotr Gęgała';
$uczen[4] = 'Cyprian Norwid';

// Tablica indeksowana 'tekstem'
$imie['Kowalski'] = 'Jan';
$imie['Nowak'] = 'Jan';
$imie['Norwid'] = 'Cyprian';
?>
```

W PHP tablicy nie definiuje się jakoś specjalnie. Po prostu inicjujemy kolejne elementy. Indeksy tablicy nie muszą być po kolei. Mogą też nimi być łańcuchy znaków.

Podstawowe funkcje – liczby losowe

```php
<?php

// Liczba losowa całkowita od 1 do 6 (włącznie)
$rzut_kostka = rand (1, 6);
echo "Wylosowano liczbę $rzut_kostka!";

?>
```

Podstawowe funkcje – data i czas

```php
<?php

// Zwraca czas w formacie Unixowym
$czas = time();
echo "Od 1.01.1970 r. minęło już $czas sekund!";

// Oblicza czas Unixowy odpowiadający podanej dacie
$kiedys = mktime ($godzina, $minuta, $sekunda, $miesiąc, $dzień,
$rok);

echo "Wskazana chwila w czasie nastąpi za ";
echo $kiedys-$czas;     // można odejmować czasy Unixowe
echo ' sekund!';

// Formatowanie daty/czasu
echo "Dziś jest ".date('d.m.Y', $czas);

?>
```

Pełną listę znaków formatujących do funkcji date() znaleźć można np.:
http://php.net/manual/pl/function.date.php
http://pl.wikibooks.org/wiki/PHP/Data_i_czas

HTML+PHP #3
Podstawy PHP

Sterowanie wykonaniem skryptu

Instrukcje warunkowe, pętle, $_GET

Dr inż. Małgorzata Janik

Instrukcja if

```
if (warunek)
    instrukcja;

if (warunek)
{
    instrukcja1;
    instrukcja2;
    …
    instrukcjan;
}

// Przykład:
if ($c != 0)
    $wynik = $a / $c;
```

Instrukcja(e) zostanie wykonana, jeśli warunek jest spełniony (tzn. jego wartość logiczna to true),

Instrukcja if-else

```
if (warunek)
    instrukcja_prawda;
else
    instrukcja_falsz;

if (warunek)
{
    instrukcja_prawda1;
    instrukcja_prawda2;
    …
    instrukcja_prawdan;
}
else
{
    instrukcja_falsz1;
    instrukcja_falsz2;
    …
    instrukcja_falszn;
}
```

```
<?php

// Przykład:
if ($c != 0)
    $wynik = $a / $c;
else
{
    echo "Głupi jesteś!\n";
    echo 'Dzielenie przez 0!';

}

?>
```

Instrukcja_prawda zostanie wykonana, jeśli warunek jest spełniony; w przeciwnym wypadku zostanie wykonana *instrukcja_falsz*.

Instrukcja if-else if-else

```
if (warunek1)
    instrukcja_prawda1;
else if (warunek2)
    instrukcja_prawda2;
else if (warunek3)
    instrukcja_prawda3;
else
    instrukcja_falsz;
```

```
<?php
// Przykład:
$x = -4;
if ($x > 0)
    echo 'Liczba dodatnia';
else if ($x < 0)
    echo 'Liczba ujemna';
else
    echo 'Zero!';
?>
```

Instrukcja_prawda1 zostanie wykonana, jeśli warunek1 jest spełniony; w przeciwnym wypadku, jeśli warunek2 jest spełniony, zostanie wykonana *instrukcja_prawda2*, itd. Ostatecznie, jeśli żaden z warunków nie został spełniony, zostanie wykonana *instrukcja_falsz*.

Ilość występujących po sobie instrukcji else if jest dowolna; ostatnia instrukcja else (oczywiście razem z instrukcja_falsz) jest opcjonalna. Każdą z pojedynczych instrukcji_prawda lub instrukcja_falsz można zastąpić blokiem instrukcji.

Petla *for*

```
for (wyr_start; warunek; inkr)
    instrukcja;

for (wyr_start; warunek; inkr)
{
    instrukcja1;
    instrukcja2;
    …
    instrukcjan;
}
```

```
<?php
// Przykład – suma ciągu 1…10:
$suma = 0;
for ($n = 1; $n <= 10; $n++)
    $suma += $n;
// $suma == 55


// Przykład – 20 razy:
for ($i = 0; $i < 20; $i++)
{
    echo 'Brawo!<br>';
    echo "$i<br>:;
}
?>
```

Przed rozpoczęciem pętli wykonywane jest *wyr_start* (np. definicja tzw. iteratora); następnie blok pętli jest wykonywany dopóki *warunek* jest prawdziwy, przy czym po każdym wykonaniu bloku wykonywana jest instrukcja *inkr* (najczęściej inkrementacja iteratora).

Petla *for* – przykłady

```php
<?php

// Przykład #4 – kolejne potęgi „dwójki” nie większe od 1024
for ($liczba = 1; $liczba <= 1024; $liczba *= 2)
    echo "$liczba\n";


// Przykład #5 – ciąg arytmetyczny od 1 do 1001 z różnicą 4
for ($liczba = 1; $liczba <= 1001; $liczba += 4)
    echo "$liczba ";


// Przykład #6 – kwadraty liczb naturalnych do 10
for ($liczba = 1; $liczba <= 10; $liczba++)
{
    echo $liczba*$liczba;
    echo ' ';
}

?>
```

Parametry GET

Przykładowy URL:
http://lubie.to/moj_plik.php?zmienna1=wartość1&zmienna2=wartość2

```php
<?php

// Pobranie wartości zmiennych z URL
// za pomocą superglobalnej tablicy $_GET[]
$z1 = $_GET['zmienna1'];
$z2 = $_GET['zmienna2'];

echo $z1;   // wyświetli wartość1
echo $z2;   // wyświetli wartość2

?>
```

Umieszczanie formularza

```
<form action="skrypt.php" method= ”get”>

<p><label>Imię i nazwisko: </label>
<input type="text" size="30" maxlength="30" name="nazwisko" /></p>

<p><input type="submit" value=" Wyślij " /><p>

</form>
```

Po wciśnięciu przycisku „Wyślij” nastąpi przekierowanie do skrypu skrypt.php z parametrami wpisanymi do formularza.
Alternatywną metodą wysyłania danych jest metoda ”post”

Young Programmer: HTML+PHP

fundacja INCEPTIO

Powodzenia!

Treść zadania do zrealizowania na pierwszych zajęciach znajduje się na stronie:

http://www.if.pw.edu.pl/~majanik/wiki/index.php/HTML%2BPHP/Zadanie3

Dr inż. Małgorzata Janik

Petla *for* – przykład

```php
<?php

// Przykład #3 – obliczanie silni
$n = 10;    // będziemy liczyć 10!

if ($n >= 0)
{
    $silnia = 1;
    for ($i = 1; $i <= $n; $i++)
        $silnia *= $i;
    echo "$n! wynosi $silnia";
}
else
{
    echo 'Nie mogę obliczyć silni z liczby ujemnej!';
}

?>
```