

Laboratorium 2

Instrukcja warunkowa

Instrukcja warunkowa w PHP ma następującą formę:

```
if (warunek1) {  
    blok instrukcji 1  
} elseif (warunek2) {  
    blok instrukcji 2  
} elseif (warunek3) {  
    blok instrukcji 3  
...  
} else {  
    blok instrukcji n  
}
```

lub formę uproszczoną:

```
if (warunek) {  
    blok instrukcji 1  
}
```

Oto przykład wykorzystania instrukcji warunkowej:

```
if ($i>0) {  
    print ('zmienna i jest większa od 0 ' )  
}  
else {  
    print ('zmienna i jest mniejsza lub równa 0')}
```

Powyższy przykład można nieco rozbudować, w celu odróżnienia wartości zmiennej **\$i** równej i mniejszej od zera:

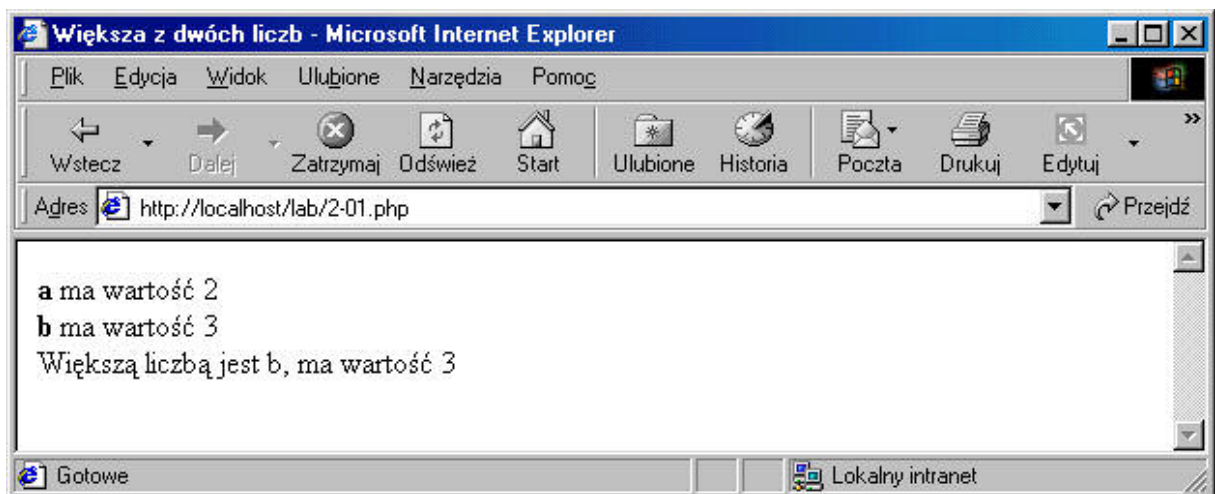
```
if ($i>0) {  
    print ('zmienna i jest większa od 0 ' )  
} elseif ($i ==0) {  
    print ('zmienna i jest równa 0')} else {  
    print ('zmienna i jest mniejsza lub równa 0')}
```

Ćwiczenie 2.1.

Napisz skrypt, który wypisze większą wartość z dwóch określonych liczb.

```
//Listing 2-1.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Większa z dwóch liczb</TITLE>
</HEAD>
<BODY>
<?
// W instrukcji if sprawdzamy, czy większe jest $a czy $b.
$a = 2;
$b = 3;
print("<B>a</B> ma wartość $a<BR>");
print("<B>b</B> ma wartość $b<BR>");
if ($a>$b) {
print ("Większą liczbą jest a, ma wartość $a");
} elseif ($a==$b) {
print ("Liczby a i b są sobie równe, obie mają wartość $a");
} else {
print ("Większą liczbą jest b, ma wartość $b");
}
?>
</BODY>
</HTML>
```

Poniżej znajduje się efekt działania skryptu.



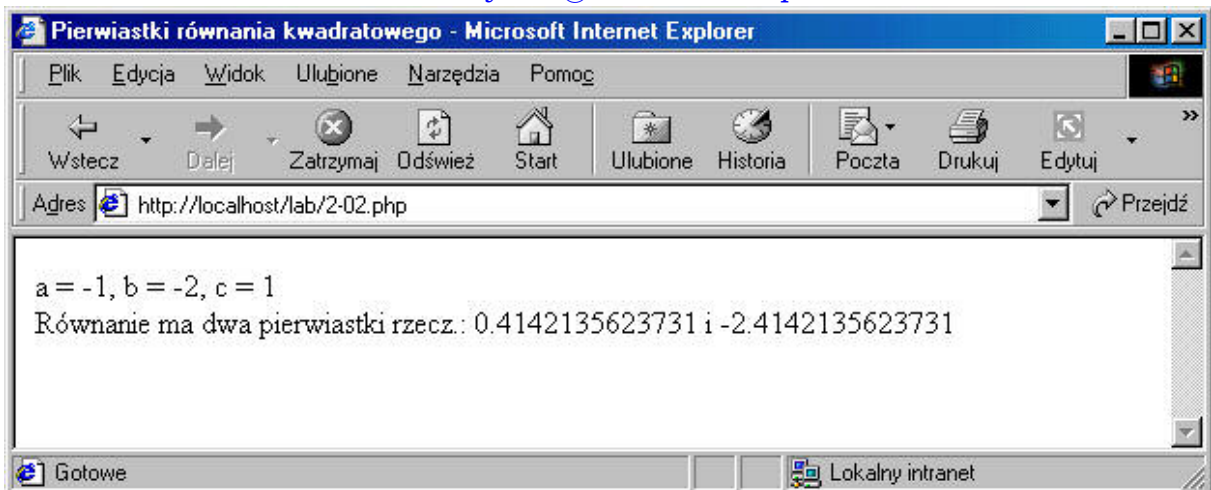
Rys. 2.1. Wynik działania skryptu.

Ćwiczenie 2.2

Wylicz pierwiastki równania kwadratowego o współczynnikach A , B i C .

Aby obliczyć pierwiastki równania kwadratowego najpierw należy wyliczyć **delte**: B^2-4AC . W zależności od jej znaku, równanie posiada zero, jedno lub dwa rozwiązania.

```
//Listing 2-2.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Pierwiastki równania kwadratowego</TITLE>
</HEAD>
<BODY>
<?
// Pierwiastki równania kwadratowego obliczamy zgodnie
// ze wzorem.
$a = -1; $b = -2; $c = 1;
print ("a = $a, b = $b, c = $c<BR>");
$delta = $b*$b-4*$a*$c;
if ($delta < 0) {
    print ('Równanie nie ma pierwiastków rzeczywistych');
}
elseif ($delta == 0) {
    $x1 = -$b/(2 * $a);
    print ("Równanie ma jeden pierwiastek rzeczywisty: $x1");
}
else {
    $x1 = (-$b-sqrt($delta)) / (2*$a);
    $x2 = (-$b+sqrt($delta)) / (2*$a);
    print ("Równanie ma dwa pierwiastki rzecz.: $x1 i $x2");
}
?>
</BODY>
</HTML>
```



Rys. 2.2. Efekt programu.

Pętla for

W PHP mamy do dyspozycji kilka możliwości „zapętlenia”, czyli wielokrotnego wykonywania bloku instrukcji. Najprostszą jest pętla **for**, która ma postać:

```
for (ustalenie_poczatku; warunek_petli; zwiększenie_licznika)
{
    blok_instrukcji
}
```

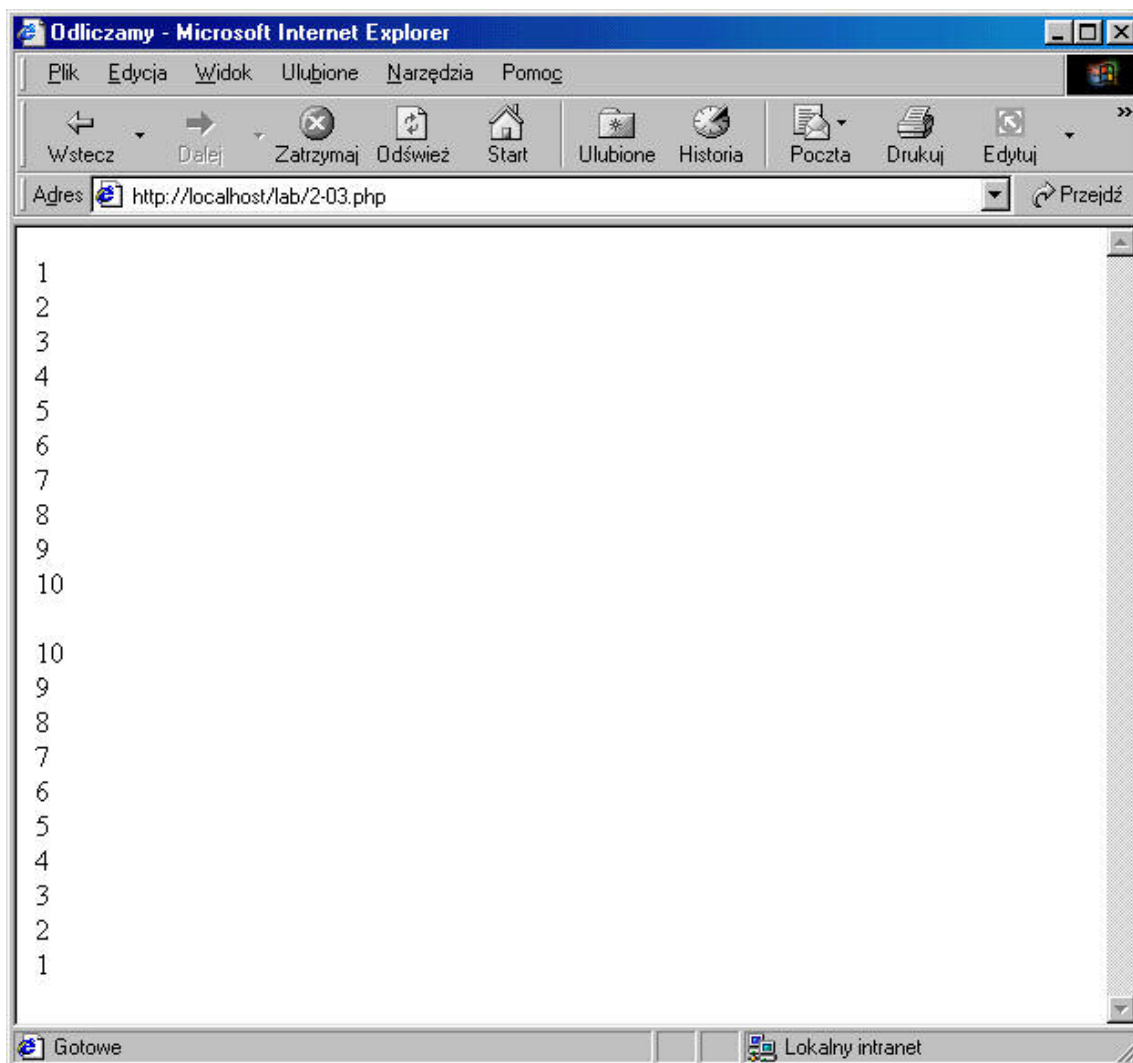
W sekcji pierwszej (**ustalenie początku**) nadajemy zmiennej sterującej w pętli wartość początkową. Druga sekcja (**warunek pętli**) to wyrażenie określające warunek, przy którego spełnieniu pętla ma być wykonywana. W trzeciej sekcji (**zwiększanie licznika**) określamy sposób zwiększania zmiennej sterującej przy każdym przebiegu pętli.

Ćwiczenie 2.3

Napisz program, który wypisze liczby od 1 do 10, a następnie od 10 do 1.

```
//Listing 2-3.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Odliczamy</TITLE>
</HEAD>
<BODY>
<?
    // Wykorzystujemy dwie pętle - "w górę" i "w dół".
    for ($i=1; $i<=10; $i++) {
        print ("{$i}<BR>") ;
    }
    print ( '<BR>' ) ;
    for ($i=10; $i>=1; $i-- ) {
        print ("{$i}<BR>");
    }
}
```

```
}  
?>  
</BODY>  
</HTML>
```



Rys. 2.3. Efekt działania programu.

Pętle while i do...while

Pętla **while** ma postać:

```
while (warunek)  
{  
    blok instrukcji  
}
```

Pętla **do ...while** ma postać:

```
do {  
    blok instrukcji  
}
```

```
while (warunek)
```

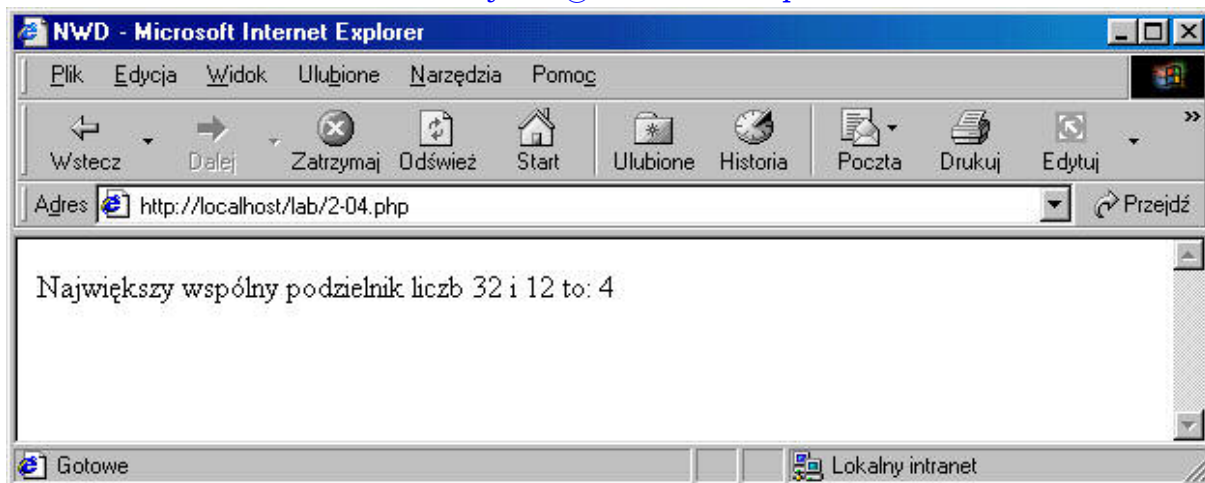
Istotę tych instrukcji dobrze oddaje ich przetłumaczenie na język polski:

1. Dopóki jest spełniony warunek, wykonuj instrukcje.
2. Wykonuj instrukcje, dopóki jest spełniony warunek.

Ćwiczenie 2.4

Napisz program, który znajdzie najmniejszy wspólny dzielnik dwóch liczb: A i B.

```
//Listing 2-4.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>NWD</TITLE>
</HEAD>
<BODY>
<?
// Największy wspólny dzielnik liczb a i b znajdziemy
// stosując odmianę algorytmu Euklidesa.
$a = 32; $b = 12;
print ("Największy wspólny dzielnik liczb $a i $b to: ");
while ($a != $b) {
    if ($a < $b){
        $pora = $a; $a = $b; $b = $pora;
    }
    $a = $a - $b;
}
print ("$a");
?>
</BODY>
</HTML>
```



Rys. 2.4. Efekt programu w przeglądarce.

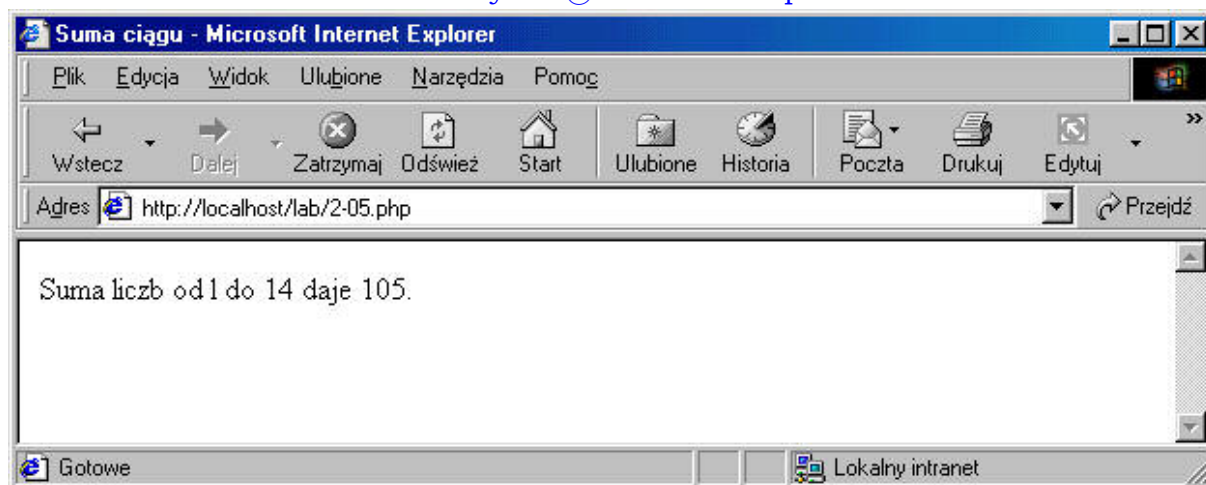
Ćwiczenie 2.5

Napisz program, który sprawdzi, ile kolejnych liczb naturalnych, zaczynających się od 1, daje sumę większą od 100.

Do napisania skryptu wykorzystamy funkcję **do..while**, w której będą sumowane kolejne wyrazy ciągu, aż do uzyskania sumy 100.

```
//Listing 2-5.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Suma ciągu</TITLE>
</HEAD>
<BODY>
<?
// Szukamy najmniejszego i takiego, że suma liczb od
// 1 do i przekracza 100.
$i = 0;
$suma = 0;
do {
    $suma += ++$i;
}
while ($suma < 100);
print ("Suma liczb od 1 do $i daje $suma.");
?>
</BODY>
</HTML>
```

INSTYTUT TECHNIKI
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI
mdymek@univ.rzeszow.pl



Rys. 2.5. Efekt programu.

Instrukcja wyboru [switch]

Instrukcja wyboru **switch** umożliwia rozgałęzienie działania programu w zależności od wartości wyrażenia. Oto postać tej instrukcji:

```
switch (wyrażenie)
{
    case wartosc1:
        dzialanie1;
        break;
    case wartosc2:
        dzialanie2;
        break;
    ...
    default dzialanie;
}
```

Wartości **wartość1**, **wartosc2**, ... (zwane wyrażeniami wyboru) służą do określenia, które działanie ma zostać podjęte (to, dla którego warunek odpowiada wartości). W przypadku, gdy nie jest to spełnione dla żadnej wartości, wykonywane są instrukcje z bloku **default**.

Ćwiczenie 2.6

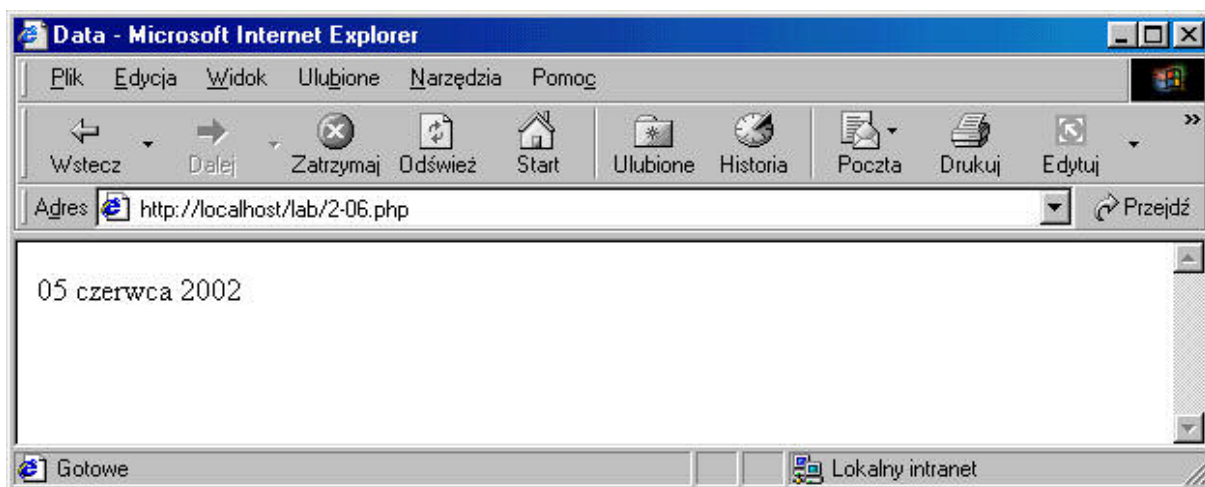
Napisz program, który wypisze bieżącą datę z miesiącem w języku polskim.

```
//Listing 2-6.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Data</TITLE>
</HEAD>
<BODY>
<?
// Polska nazwa miesiąca określona w funkcji switch
// do uzyskania dnia, miesiąca i roku używamy funkcji date.
$dzien = date("d");
$miesiac = date("m");
$rok = date("Y");
switch ($miesiac) {
    case '01': $miesiac = 'stycznia'; break;
    case '02': $miesiac = 'lutego'; break;
    case '03': $miesiac = 'marca'; break;
    case '04': $miesiac = 'kwietnia'; break;
    case '05': $miesiac = 'maja'; break;
    case '06': $miesiac = 'czerwca'; break;
    case '07': $miesiac = 'lipca'; break;
    case '08': $miesiac = 'sierpnia'; break;
```

**INSTYTUT TECHNIKI
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

mdymek@univ.rzeszow.pl

```
case '09': $miesiac = 'wrzeźnia'; break;
case '10': $miesiac = 'października'; break;
case '11': $miesiac = 'listopada'; break;
case '12': $miesiac = 'grudnia'; break;
default: $miesiac = 'niezidentyfikowany'; break;
}
print "$dzien $miesiac $rok";
?>
</BODY>
</HTML>
```



Rys. 2.6. Efekt programu.

Podając format daty można użyć następujących znaków specjalnych:

Znak	Opis
D	dzień tygodnia w skróconym zapisie tekstowym, na przykład "Fri"
I	dzień tygodnia w zapisie tekstowym, na przykład "Monday"
w	dzień tygodnia w postaci numerycznej od "0" (Niedziela) do "6" (Sobota)
z	dzień w roku od "0" do "365"
d	dzień miesiąca w zapisie numerycznym od "01" do "31"
F	miesiąc w zapisie tekstowym np.: "January"
M	miesiąc w skróconym zapisie tekstowym, na przykład "Jan"
m	miesiąc w zapisie numerycznym od "01" do "12"
t	liczba dni w miesiącu od "28" do "31"
h	godzina, format 12-godzinowy od "01" do "12"
H	godzina, format 24-godzinowy od "00" do "23"
a	"am", czy "pm"
A	"AM", czy "PM"
i	minuty, od "00" do "59"
s	sekundy od "00" do "59"
Y	rok w zapisie 4-cyfrowym, na przykład "2002"
y	rok w zapisie 2-cyfrowym, na przykład "02"

Zadania

1. Dobierz takie wartości zmiennych \$a\$, \$b\$ i \$c\$ w ćwiczeniu 2.2., by uzyskać równanie bez, z jednym i z dwoma rozwiązaniami rzeczywistymi.
2. Napisz skrypt, który wyświetli powitanie wraz z aktualną datą.
3. Napisz program, który obliczy i wyświetli $N!$.
4. Napisz program, który sprawdzi, czy n jest liczbą pierwszą.