

Laboratorium 3

Funkcje

Definicja funkcji musi nastąpić przed jej wywołaniem i ma postać:

```
function nazwa (lista_parametrów)
{
    blok instrukcji
}
```

Parametry, czyli dane przekazane do funkcji, są widoczne wewnątrz niej pod odpowiednimi nazwami (takimi, jakie zadeklarowaliśmy w nagłówku funkcji) i można na nich operować tak jak na zmiennych.

Jeżeli chcesz, by funkcja zwróciła jakąś wartość, należy posłużyć się instrukcją **return**. Powoduje ona zakończenie działania funkcji i zwrócenie jako wyniku wartości wyrażenia występującego w instrukcji **return**.

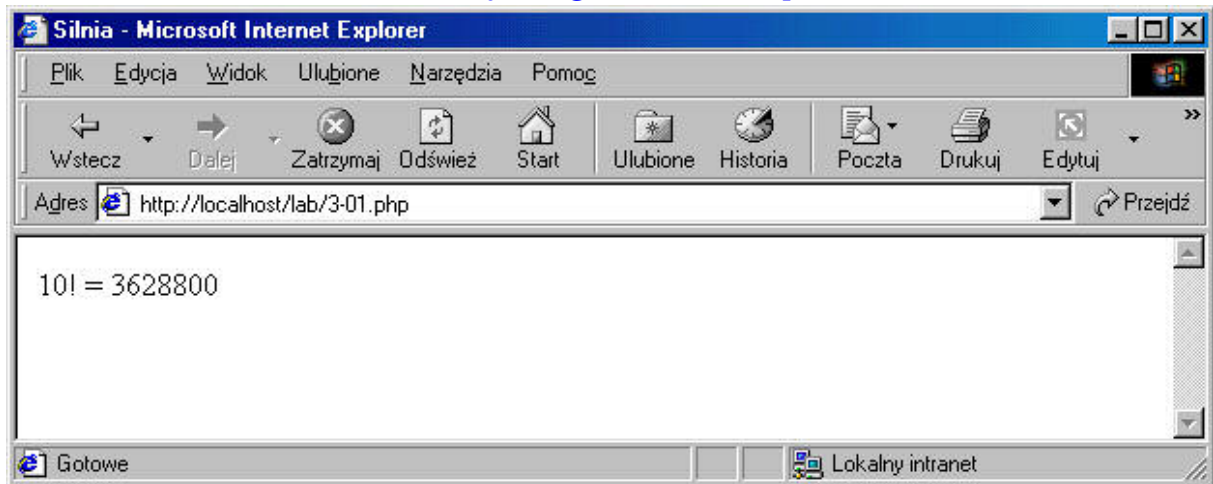
Ćwiczenie 3.1.

Napisz funkcję, która oblicza silnię liczby, i program sprawdzający jej działanie.

Zwróć uwagę, że do napisanej już funkcji możesz się odwołać tak, jak do standardowej.

```
//Listing 3-1.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Silnia</TITLE>
</HEAD>
<BODY>
<?
    function silnia ($n) {
        $silnia = 1;
        for ($i=1; $i<=$n; $i++) {
            $silnia *= $i;
        }
        return $silnia;
    }
    print ("10! = ".silnia(10));
?>
</BODY>
</HTML>
```

Zmienne **\$i** oraz **\$silnia**, używane w funkcji **silnia** są na zewnątrz niewidoczne. Noszą one nazwę **zmiennych lokalnych funkcji**. Z kolei taka, która jest używana w głównym bloku programu, to **zmienna globalna**. Istnieje możliwość użycia w funkcji zmiennych lokalnych o takich samych nazwach, jak zmienne globalne.



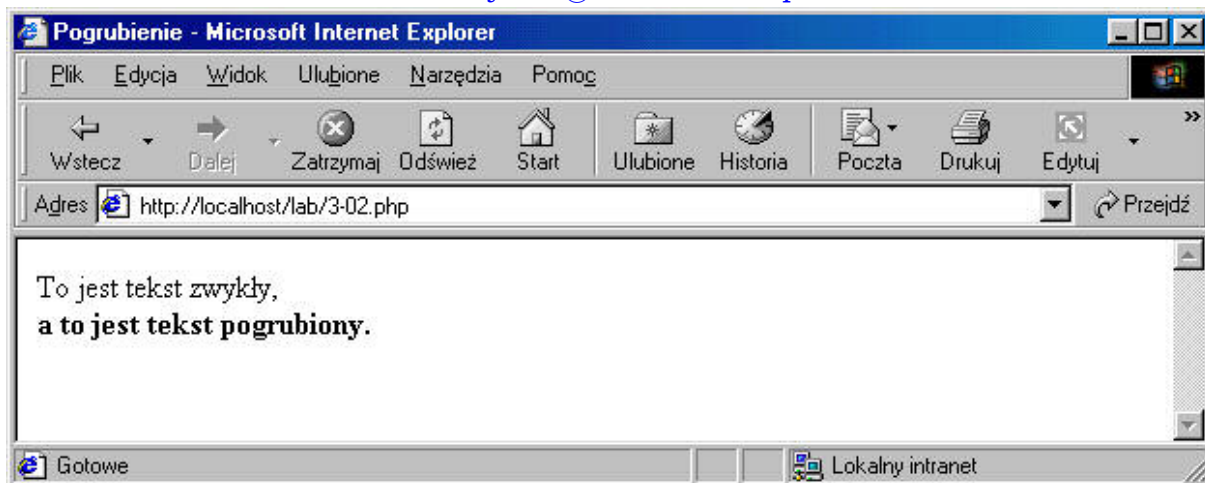
Rys. 3.1. Wynik działania skryptu.

Jeżeli chciałbyś skorzystać ze zmiennej globalnej, możesz użyć instrukcji **global**. Powoduje ona, że na zmiennej będzie można operować tak samo, jak na lokalnej, a jeżeli funkcja zmieni jej wartość, po zakończeniu działania pozostanie ona zmieniona.

Ćwiczenie 3.2.

Napisz funkcję, która łańcuch sformatuje pogrubieniem.

```
//Listing 3-2.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Pogrubienie</TITLE>
</HEAD>
<BODY>
<?
function pogrubienie ($lancuch)
{
    return "<B>".$lancuch."</B>";
}
print ("To jest tekst zwykły,<BR>");
print pogrubienie("a to jest tekst pogrubiony.");
?>
</BODY>
</HTML>
```



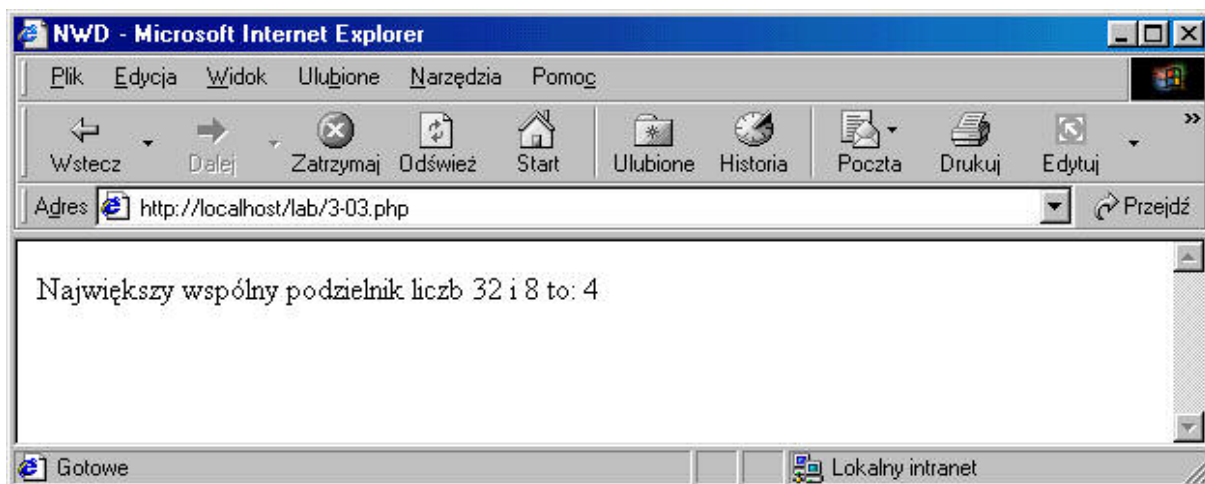
Rys. 3.2. Efekt działania funkcji.

Ćwiczenie 3.3.

Napisz funkcję, która oblicza największy wspólny dzielnik dwóch liczb, i program sprawdzający jej działanie.

```
//Listing 3-3.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>NWD</TITLE>
</HEAD>
<BODY>
<?
function nwd ($a, $b)
{
    // Funkcja znajduje największy wspólny podzielnik liczb
    // a i b znajdziemy stosując odmianę algorytmu Euklidesa.
    $a = 32;
    $b = 12;
    while ($a != $b) ( if ($a < $b)
    {
        $pora = $a;
        $a = $b;
        $b = $pom;
    }
    $a = $a - $b;
}
return $a;
}
$a = 32;
```

```
$b = 8;  
print ("Największy wspólny dzielnik liczb $a i $b to: ");  
print nwd ($a, $b) ;  
?>  
</BODY>  
</HTML>
```



Rys. 3.3. Wynik działania funkcji.

Ćwiczenie 3.4.

Napisz funkcję, która dla parametru będącego imieniem, zgaduje płeć osoby.

Zgadywanie niestety nie będzie skuteczne w 100%. Metoda będzie polegać na tym, że jeżeli ostatnią literą imienia będzie „a”, ocenimy, że mamy do czynienia z kobietą, jeżeli nie - z mężczyzną.

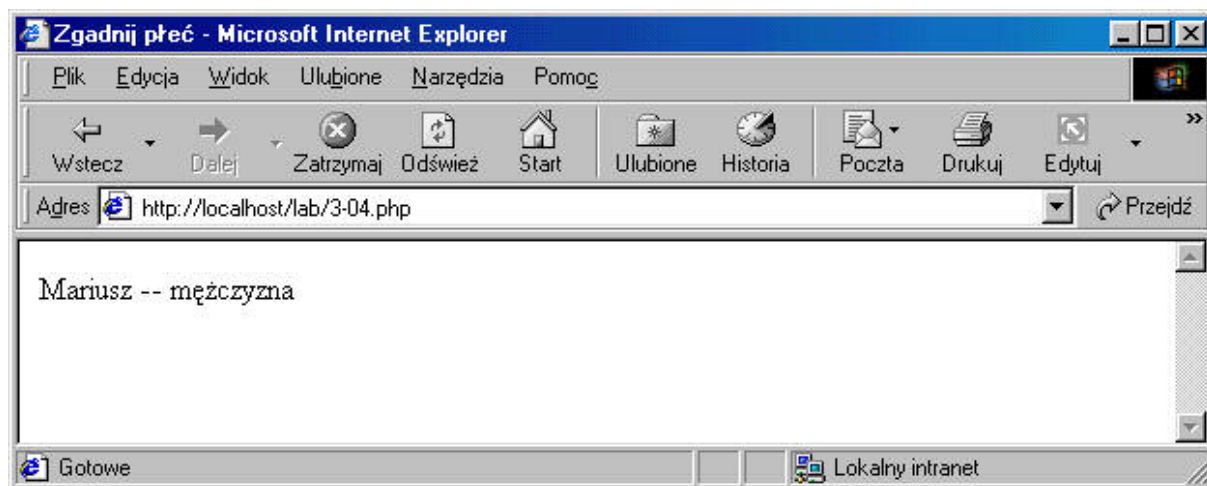
Aby zrealizować ćwiczenie należy skorzystać z dwóch predefiniowanych funkcji. Funkcja **substr** pozwoli nam uzyskać dowolny podciąg tekstu. Aby jednak otrzymać ostatnią literę, musisz też znać jego długość. W tym celu wykorzystaj funkcję **strlen**.

```
//Listing 3-4.php  
<HTML>  
<HEAD>  
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">  
<TITLE>NWD</TITLE>  
</HEAD>  
<BODY>  
<?>  
function plec ($imie)  
{  
    // Funkcja próbuje zgadnąć płeć osoby na podstawie  
    // ostatniej litery jej imienia.  
    $ostatnialitera = substr($imie, strlen($imie)-1, 1);  
    if ($ostatnialitera =='a') { return 'kobieta'; }  
}
```

**INSTYTUT TECHNIKI
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

mdymek@univ.rzeszow.pl

```
        else { return 'mężczyzna'; }  
    }  
    print ("Mariusz -- ".plec ('Mariusz'));  
?>  
</BODY>  
</HTML>
```



Rys.3.4. Efekt działania skryptu.

Najczęściej używane gotowe funkcje:

substr	zwraca podłańcuch pierwszego parametru, zaczynający się od znaku o numerze określonym drugim parametrem i długości określonej trzecim. Należy pamiętać, że znaki są liczone od 0
strlen	zwraca długości łańcucha określonego parametrem
require	włączenie pliku - argumentu do dokumentu
include	wykonanie instrukcji z pliku - argumentu
pow(\$a, \$b)	podnosi \$a do potęgi \$b

Zadania

1. Napisz funkcję, która oblicza $N!$ w sposób rekurencyjny.
2. Napisz funkcję, która oblicza metodą rekurencyjną naturalną potęgę (n) liczby całkowitej c , i program sprawdzający jej działanie.
3. Napisz program, który wypisuje wszystkie podzbiory zbioru liczb naturalnych mniejszych lub równych N .