

## Laboratorium 1

**PHP** jest językiem skryptowym, wykonywanym po stronie serwera, a pisane w nim skrypty są umieszczane w kodzie **HTML**.

Dokumenty zawierające skrypty PHP (pliki zazwyczaj mają **rozszerzenie .php, .php3, .php4**) to pomieszczenie kodu HTML i tekstu programu. W rzeczywistości w kodzie HTML tekst programu jest wydzielony **znacznikiem** z tego kodu (choć może też być odwrotnie: kod HTML-owy może być zawarty w skrypcie PHP!). Początek i koniec programu mogą być oznaczone odpowiednio przez:

```
<?php                                ?>
<?                                  ?>
<SCRIPT LANGUAGE="php">            </SCRIPT>
```

**Komentarz** jest fragmentem kodu pomijanym przez interpreter. Musimy oczywiście odpowiednio ów opis zaznaczyć:

- `//` - interpreter pominie wszystko co znajdzie za tymi znakami aż do końca linii
- `/* i */` - oznaczają odpowiednio początek i koniec komentarza obejmującego jedną lub więcej linii
- dla kodu HTML komentarz zaznaczamy poprzez użycie znaczników `<!--` i `-->`.

### Ćwiczenie 1.1.

**Utwórz w PHP program, który wypisze Twoje imię i nazwisko.**

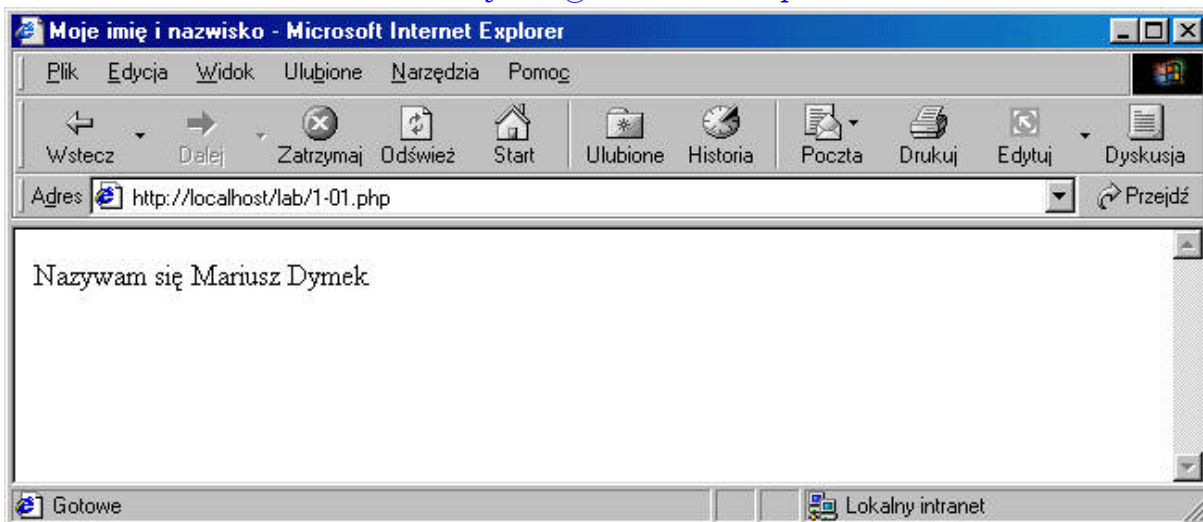
Musimy zapoznać się z funkcją, która pozwala na wypisanie tekstu.

<b>print</b>	Najprostsza funkcja wypisująca podaną wartość.
<b>echo</b>	Funkcja wypisująca podaną wartość, podobna do print.
<b>printf</b>	Funkcja pozwalająca na wypisanie wartości po jej wcześniejszym sformatowaniu. Jest bardzo przydatna, a zapoznamy się z nią w ćwiczeniu 3.5.

Napiszmy teraz program:

```
//Listing 1-1.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Moje imię i nazwisko</TITLE>
</HEAD>
<BODY>
<?
    // Wypisujemy imię i nazwisko, używając funkcji "print"
    print ("Nazywam się Mariusz Dymek");
?>
</BODY>
</HTML>
```

Po umieszczeniu pliku na serwerze (nie jest ważne, czy to obcy serwer w Internecie, gdzie mamy konto, czy też zainstalowany Apache na naszym komputerze) i wpisaniu adresu dokumentu uzyskamy wynik działania pierwszego napisanego skryptu PHP (rysunek 1.1).



Rys. 1.1. Wynik działania pierwszego skryptu PHP

## Ćwiczenie 1.2.

Utwórz skrypt, który wypisze Ci informacje o konfiguracji PHP na serwerze.

Wykorzystamy funkcję, która zwraca stronę informacyjną o konfiguracji serwera.

<b>phpinfo</b>	Funkcja zwracająca stronę informacyjną o konfiguracji PHP i o bieżącym połączeniu i zmiennych systemowych
----------------	---

Oto skrypt:

```
//Listing 1-2.php
<?
// Wykorzystujemy funkcję "phpinfo" w celu uzyskania
// informacji o konfiguracji PHP.
phpinfo();
?>
```

W przeglądarce wyświetli się strona z informacjami (rysunek 1.2).



**INSTYTUT TECHNIKI  
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

[mdymek@univ.rzeszow.pl](mailto:mdymek@univ.rzeszow.pl)

Rys. 1.2. Strona informacyjna wygenerowana za pomocą funkcji **phpinfo**.

Funkcja **phpinfo** wyświetla kompletny tekst HTML strony informacyjnej.

W sekcji **Environment** znajduje się lista zmiennych środowiskowych, np.:

- **HTTP\_HOST**, w której jest zapisana jego nazwa;
- **REQUEST\_URI**, pamiętająca nazwę skryptu;
- **SERVER\_ADMIN**, z której można odczytać adres e-mail administratora serwera;
- **REMOTE\_ADDR**, która określa adres **IP** odwiedzającego;
- **HTTP\_USER\_AGENT**, z której można wyczytać, z jakiego systemu operacyjnego korzysta i jaką posługuje się przeglądarką.

### **Zmienne, stałe, operatory.**

Nazwy zmiennych mogą składać się z liter, cyfr i znaków podkreślenia, jednak nie mogą zaczynać się od cyfry. W programie poprzedza się je znakiem dolara (\$). W PHP nie jest wymagane wcześniejsze deklarowanie ich przed użyciem, a więc wcześniejsze określanie jej typu

Zmienne w PHP dzielą się na typy, np.:

- łańcuchy znakowe;
- liczby całkowite;
- liczby zmiennopozycyjne.

### **Ćwiczenie 1.3.**

**Utwórz zmienne, w których zapamiętasz swoje imię i nazwisko i wypisz je.**

Skrypt realizujący to zadanie:

```
//Listing 1-3.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Moje imię i nazwisko</TITLE>
</HEAD>
<BODY>
<?
    // Imię i nazwisko, zapamiętujemy w zmiennych
    // a następnie wypisujemy funkcją print.
    $imie = 'Mariusz';
    $nazwisko = "Dymek";
    print ("Nazywam się " . $imie . " " . $nazwisko);
?>
</BODY>
</HTML>
```

Ważnym elementem są znaki kropki w instrukcji **print**, pomiędzy poszczególnymi łańcuchami. Znak kropki jest operatorem połączenia dwóch łańcuchów.

**Łańcuch tekstowy** może być zawarty w pojedynczych lub podwójnych cudzysłowach.

" ... "	W łańcuchach w podwójnych cudzysłowach następuje tak zwana <b>interpolacja zmiennych</b> .
---------	--

**INSTYTUT TECHNIKI  
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

[mdymek@univ.rzeszow.pl](mailto:mdymek@univ.rzeszow.pl)

	Oznacza to, że nazwy zmiennych zawartych w takich łańcuchach są zastępowane przez ich wartości.
'...'	W łańcuchach w pojedynczych cudzysłowach taka zmiana nie następuje.

Różnice w użyciu podwójnych i pojedynczych cudzysłowów.

```
$tekst = "Nazywam się $imie $nazwisko";
```

Nazywam się Mariusz Dymek

```
$tekst = 'Nazywam się $imie $nazwisko';
```

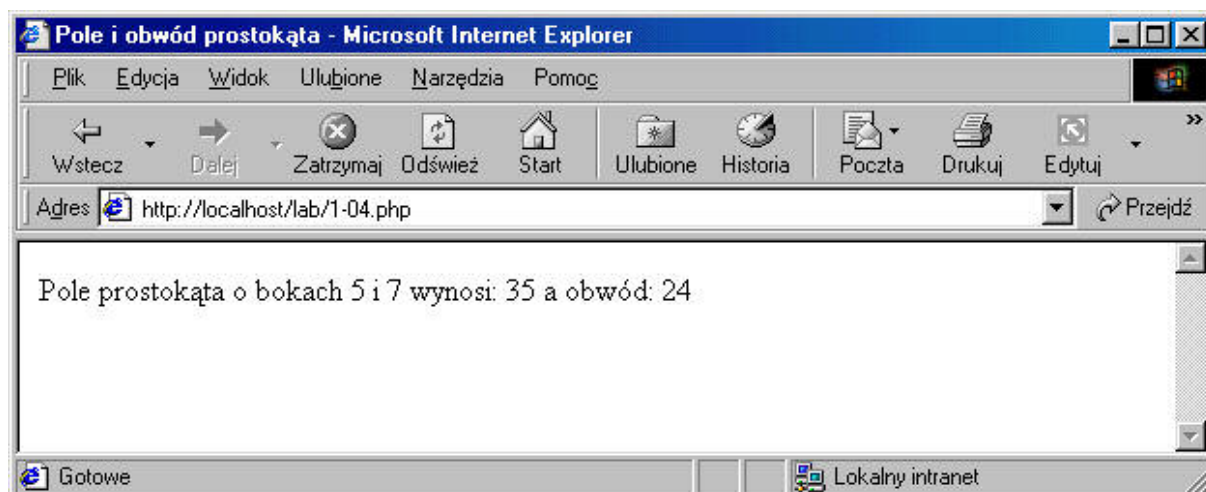
Nazywam się \$imie \$nazwisko

### **Ćwiczenie 1.4.**

**Napisz skrypt, który wypisze obwód i pole prostokąta o zadanych bokach.**

```
//Listing 1-4.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Pole i obwód prostokąta</TITLE>
</HEAD>
<BODY>
<?
    // W zmiennych zapamiętamy boki prostokąta, a
    // następnie obliczymy pole i obwód kwadratu.
    $bokA = 5;
    $bokB = 7;
    print ("Pole prostokąta o bokach $bokA i $bokB
    wynosi:". $bokA*$bokB. " a obwód: " . 2*($bokA+$bokB) );
?>
</BODY>
</HTML>
```

Efekt w przeglądarce widać na rysunku 1.3.



Rys. 1.3. Wyniki obliczeń programu.

**Stałymi** nazywamy pewne obiekty, które mają z góry określoną wartość, która nie ulega zmianie podczas wykonywania programu. Jako stałe możesz definiować nie tylko liczby, ale także teksty. Musisz je wtedy zawrzeć w cudzysłowach.

Stałe definiuje się za pomocą funkcji **define**.

<b>define</b>	Definiowanie wartości stałej.
---------------	-------------------------------

Oto przykład definicji:

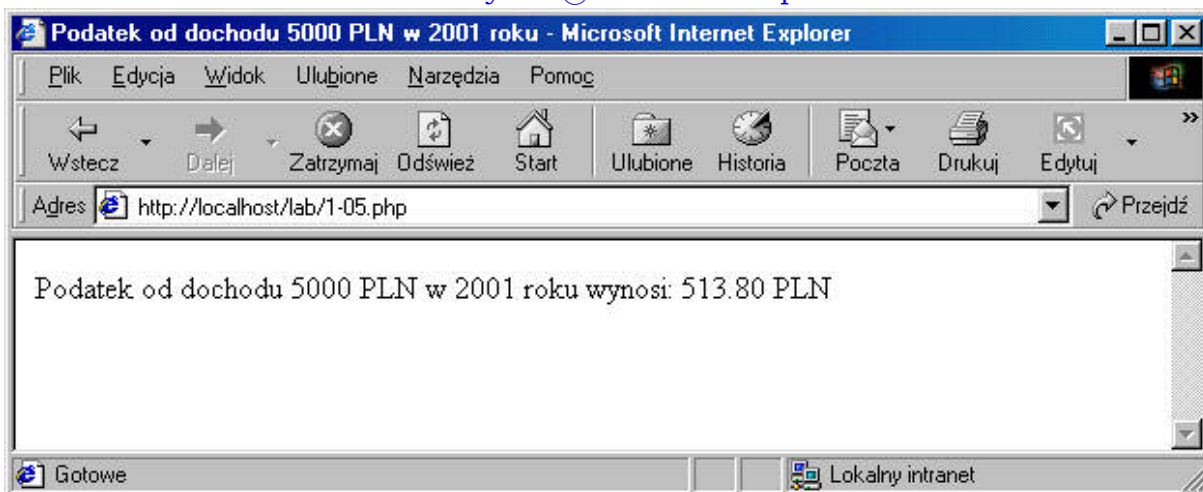
```
define ("STAWKA_PODATKOWA", 0.19)
```

Funkcja posiada dwa argumenty: **nazwę stałej** i jej **wartość**.

## Ćwiczenie 1.5

Napisz skrypt, który wyświetli wartość podatku od dochodu 5000 zł w 2001 roku. Użyj stałych do zapamiętania kwoty wolnej od podatku i stawki procentowej. Podatek należy obliczyć w następujący sposób: od dochodu trzeba odjąć kwotę wolną od podatku i następnie wynik pomnożyć przez stawkę procentową.

```
//Listing 1-5.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Podatek od dochodu 5000 PLN w 2001 roku </TITLE>
</HEAD>
<BODY>
<?
define ("KWOTA_DO_ODJECIA", 436.20);
define ("STAWKA_PODATKOWA", 0.19);
// czyli 19%
print ("Podatek od dochodu 5000 PLN w 2001 roku wynosi: ");
printf ("%0.2f", (5000*STAWKA_PODATKOWA - KWOTA_DO_ODJECIA));
print (' PLN');
?>
</BODY>
</HTML>
```



Rys. 1.4. Wynik działania programu po obliczeniu podatku.

Funkcją **printf** pozwala na formatowanie wypisywanego tekstu. Funkcja wymaga argumentu, określającego sposób formatowania, po którym wpisuje się informacje do wyświetlenia, oddzielając je przecinkami.

Poniżej znajduje się lista operatorów arytmetycznych i operatorów przypisania.

#### **Operatory arytmetyczne**

Operator	Opis
+	dodawanie
-	odejmowanie
*	mnożenie
/	dzielenie
%	reszta z dzielenia

#### **Operatory przypisania**

Operator	Opis
=	liczba stojąca po lewej stronie operatora przyjmuje wartość liczby stojącej po prawej stronie operatora
+=	liczba stojąca po lewej stronie operatora przyjmuje wartość będącą sumą jej samej i liczby stojącej po prawej stronie operatora
-=	liczba stojąca po lewej stronie operatora przyjmuje wartość będącą różnicą jej samej i liczby stojącej po prawej stronie operatora
*=	liczba stojąca po lewej stronie operatora przyjmuje wartość będącą iloczynem jej samej i liczby stojącej po prawej stronie operatora
/=	liczba stojąca po lewej stronie operatora przyjmuje wartość będącą ilorazem jej samej i liczby stojącej po prawej stronie operatora
%=	liczba stojąca po lewej stronie operatora przyjmuje wartość będącą resztą z dzielenia (modulo) jej samej i liczby stojącej po prawej stronie operatora
.=	przypisuje do wyniku (będącego łańcuchem) jego połączenie z łańcuchem (będącym argumentem)
==	operator równości
!=	operator różności
!	operator zaprzeczenia (negacji), stosowany przed argumentem
++	operator zwiększenia o 1 (czyli inkrementacja)
--	operator zmniejszenia o 1 (czyli dekrementacja)

## **Ćwiczenie 1.6**

**INSTYTUT TECHNIKI  
ZAKŁAD ELEKTROTECHNIKI I INFORMATYKI**

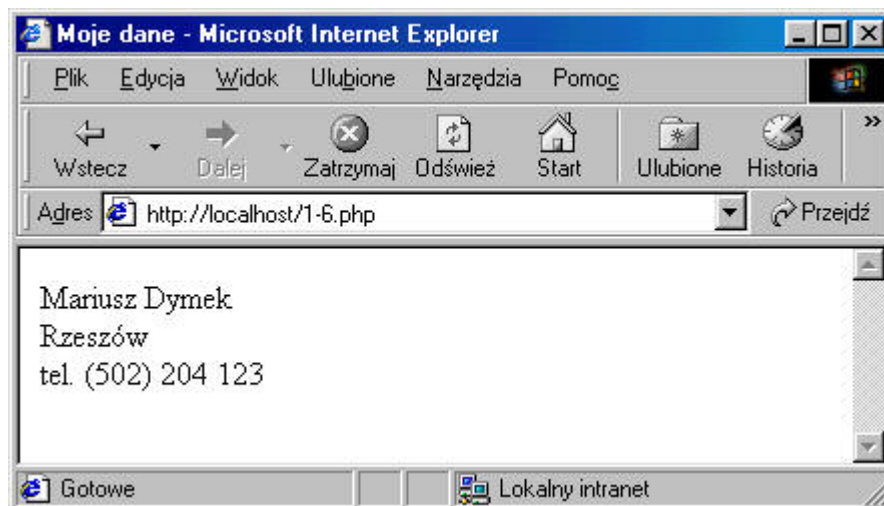
mdymek@univ.rzeszow.pl

**Napisz skrypt, który wypisze Twoje informacje adresowe, używając operatora .= (przypisującego połączenie wyniku z argumentem).**

Poszczególne informacje będziemy gromadzić w kolejnych poleceniach skryptu.

```
//Listing 1-6.php
<HTML>
<HEAD>
<META HTTP-EQUIV="Content-Type"CONTENT="text/html;charset=iso-8859-2">
<TITLE>Moje dane</TITLE>
</HEAD>
<BODY>
<?
//Kolejne informacje zostają zbierane przy użyciu operatora .=
//(przypisującego połączenie wyniku z argumentem).
    $tekst = 'Mariusz';
    $tekst .= ' Dymek';
    $tekst .= '<BR>';
    $tekst .= 'Rzeszów';
    $tekst .= '<BR>';
    $tekst .= 'tel. (502) 204 123';
    print ($tekst);
?>
</BODY>
</HTML>
```

Poniżej znajduje się wynik działania skryptu:



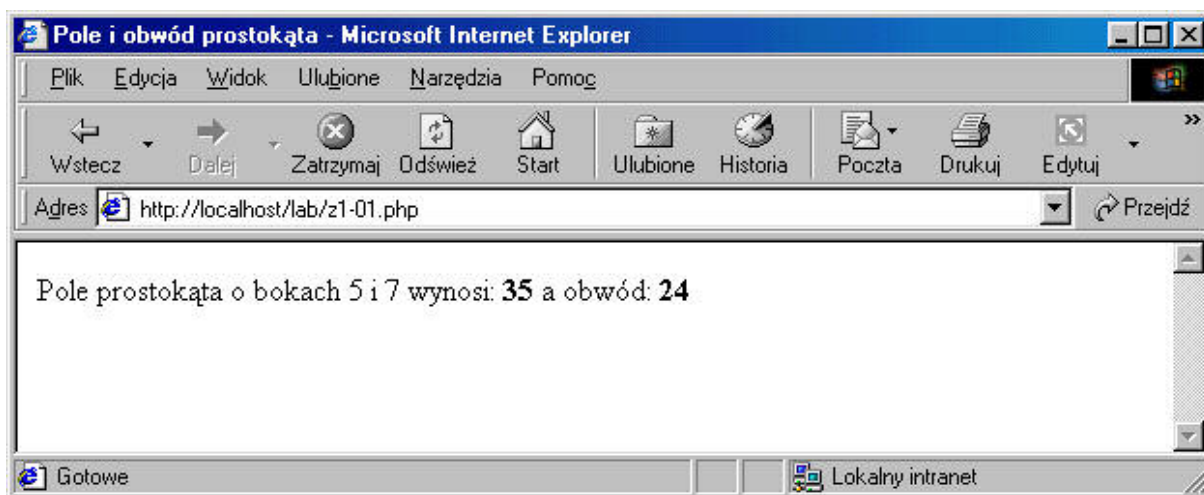
Rys. 1.5. Wynik działania programu w przeglądarce.

## **Zadania:**

---

1. Napisz skrypt, żeby wyniki obliczeń z ćwiczenia 1.4 były wyróżnione pogrubieniem.

Efekt w przeglądarce widać na rysunku 1.6.



Rys. 1.6. Wyniki obliczeń zostały wyróżnione pogrubieniem.

2. Postępując się programem z zadania 1.5. napisz program wstawiając w nim drugą instrukcję, wypisującą podatek dla dochodu 10.000 zł.
3. Napisz skrypt, który obliczy pole i obwód okręgu.