

# Napisy w PHP.

## Drukowanie napisów instrukcją echo

### 1. Ćwiczenia do samodzielnego wykonania

Wszystkie poniższe ćwiczenia należy samodzielnie wykonać wpisując w edytorze bez stosowania techniki kopiuj-wklej.

#### 1.1 Ćwiczenie

**TEMAT:** wysyłanie do klienta trzech napisów ala, ma oraz kota. Napisy otaczamy apostrofami.

```
<html>
  <head>
    <title>Ćwiczenie 1.1</title>
    <meta http-equiv="Content-Type"
      content="text/html; charset=iso-8859-2" />
  </head>
<body>

<h1>Ćwiczenie 1.1</h1>

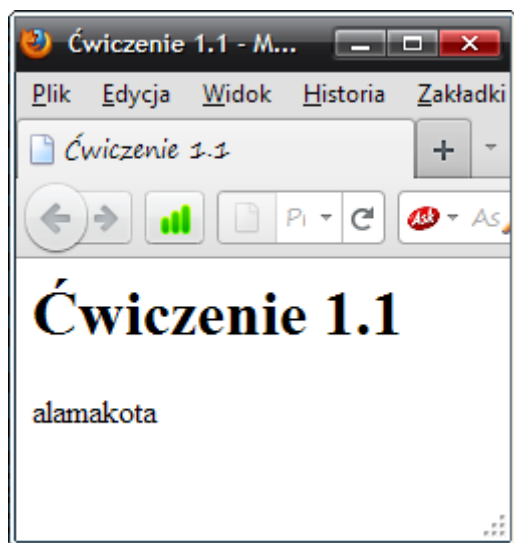
<?php

echo 'ala';
echo 'ma';
echo 'kota';

?>

</body>
</html>
```

W przeglądarce WWW powyższa strona będzie wyglądała następująco:



Zauważmy brak spacji pomiędzy wyrazami ala, ma oraz kota.

Jeśli wykonamy w przeglądarce operację Widok → Źródło, wówczas zobaczymy następujący kod HTML:

```
<html>
  <head>
    <title>Ćwiczenie 1.1</title>
    <meta http-equiv="Content-Type"
          content="text/html; charset=iso-8859-2" />
  </head>
  <body>

    <h1>Ćwiczenie 1.1</h1>

    alamakota
  </body>
</html>
```

Skrypt zawarty pomiędzy znacznikami <?php oraz ?> zostaje wykonany przez maszynę php. W jego miejsce w kodzie HTML pojawia się tekst, który został wydrukowany instrukcjami echo.

## 1.2 Ćwiczenie 1.2

**TEMAT:** napisy otaczane znakami cudzysłowu.

```
<body>

<h1>Ćwiczenie 1.2</h1>

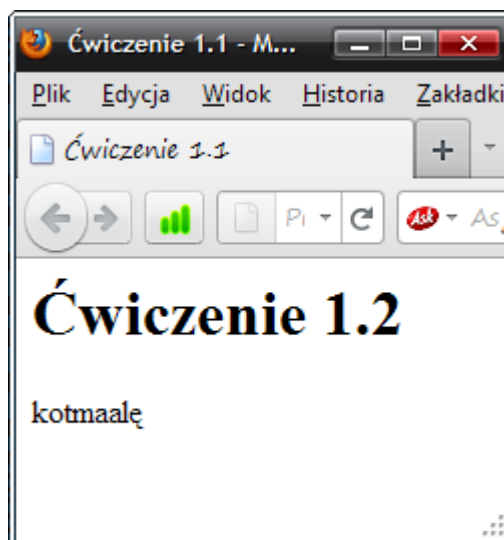
<?php

echo "kot";
echo "ma";
echo "alę";

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.2:

```
<body>

<h1>Ćwiczenie 1.2</h1>

kotmaale
</body>
```

### 1.3 Ćwiczenie 1.3

**TEMAT:** Znaki cudzysłowu i apostrofów mogą występować w jednym skrypcie. Część napisów możemy ograniczyć apostrofami, a część — znakami cudzysłowu.

```
<body>

<h1>Ćwiczenie 1.3</h1>

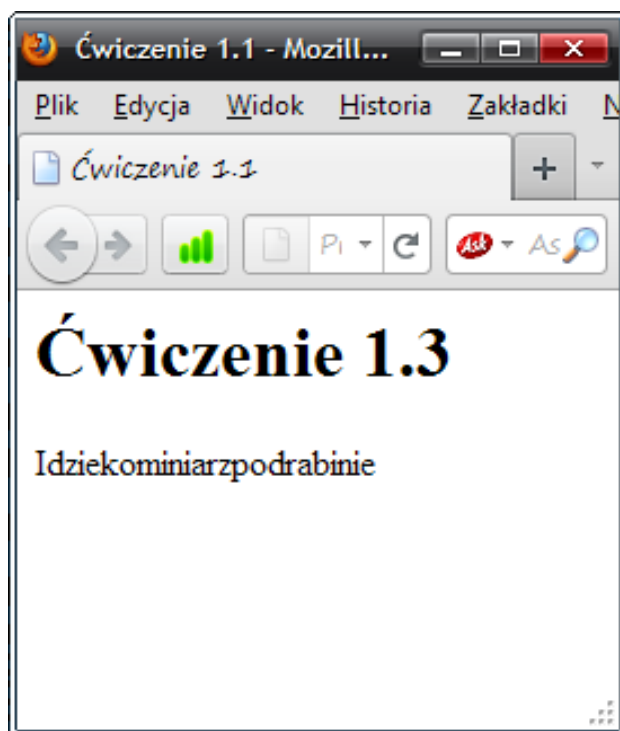
<?php

echo "Idzie";
echo 'kominiaarz';
echo "po";
echo 'drabinie';

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.3:

```
<body>

<h1>Ćwiczenie 1.3</h1>

idziekominiarzpodrabinie
</body>
```

## 1.4 Ćwiczenie 1.4

**TEMAT:** umieszczanie białych znaków w wydruku.

Wyrazy pojawiające się na stronach z poprzednich ćwiczeń zlewały się (nie były oddzielone spacją). Jeśli chcemy oddzielić wyrazy, to musimy spacje umieścić wewnątrz napisów drukowanych instrukcją echo.

```
<body>

<h1>Ćwiczenie 1.4</h1>

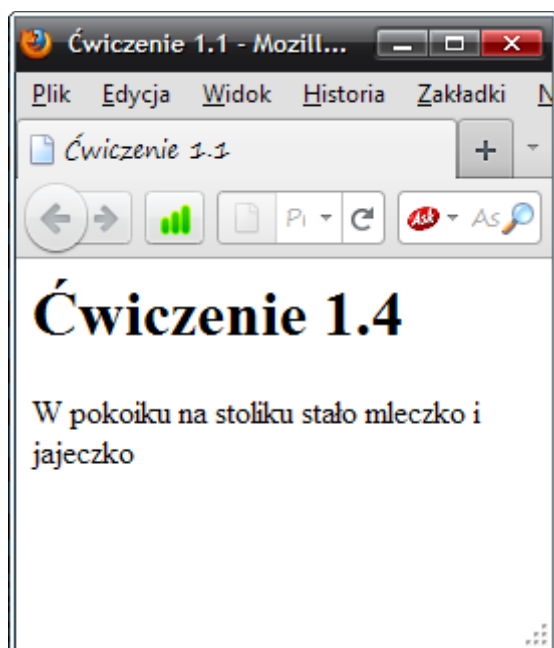
<?php

echo "W pokoiku";
echo ' na ';
echo "stoliku";
echo '      stało mleczko i      jajeczko';

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.4:

```
<body>
```

```
<h1>Ćwiczenie 1.4</h1>
```

```
W pokoiku na stoliku,          stało mleczko i          jajeczko  
</body>
```

## 1.5 Ćwiczenie 1.5

**TEMAT:** Drukowanie apostrofów oraz znaków cudzysłowu.

W celu wydrukowania znaku cudzysłowu, należy umieścić go w napisie otoczonym znakami apostrofów, np. 'Piosenka pt.'".

W celu wydrukowania znaku apostrofu należy umieścić go w napisie otoczonym znakami cudzysłowu, np. "I'll follow".

```
<body>
```

```
<h1>Ćwiczenie 1.5</h1>
```

```
<?php
```

```
echo 'Piosenka pt.'";
```

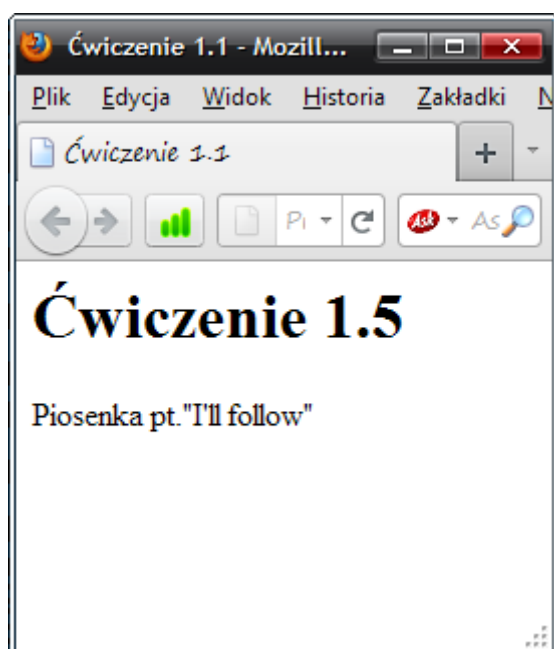
```
echo "I'll follow";
```

```
echo '";
```

```
?>
```

```
</body>
```

Wygląd skryptu z ćwiczenia 4.5 w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.5:

```
<body>

<h1>Ćwiczenie 1.5</h1>

Piosenka pt. "I'll follow"
</body>
```

## 1.6 Ćwiczenie 1.6

**TEMAT:** Drukowanie kodu HTML.

Jeśli skrypt ma generować stronę WWW w języku HTML, wówczas do klienta musimy wysłać pewną ilość znaczników HTML. Znaczniki te umieszczamy wewnątrz napisów, drukowanych instrukcją echo.

```
<body>

<h1>Ćwiczenie 1.6</h1>

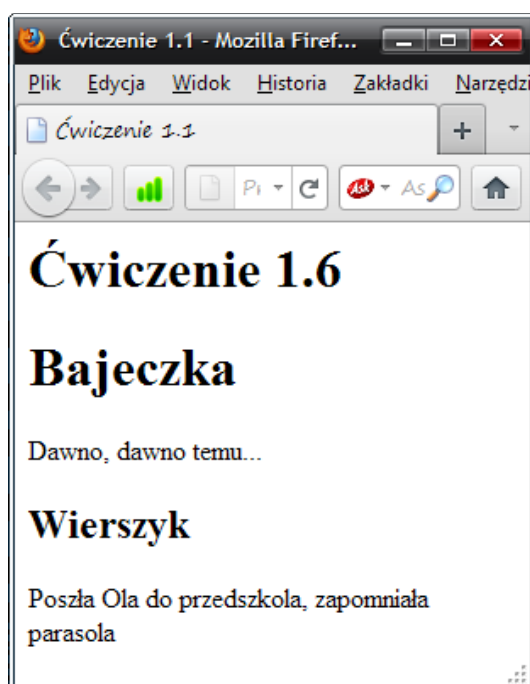
<?php

echo '<h1>Bajeczka</h1>';
echo '<p>Dawno, dawno temu...</p>';
echo "<h2>Wierszyk</h2>";
echo '<p>Poszła Ola do przedszkola, zapomniała parasola</p>';

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.6:

```
<body>

<h1>Ćwiczenie 1.6</h1>

<h1>Bajeczka</h1><p>Dawno...</p><h2>Wierszyk</h2><p>Poszła...</p>
</body>
```

## 1.7 Ćwiczenie 1.7

**TEMAT:** wysyłanie do klienta bardziej złożonego kodu HTML.

Niektóre znaczniki (np. table) posiadają atrybuty (np. border="1"). Wartości atrybutów otaczamy znakami cudzysłowu.

```
<body>

<h1>Ćwiczenie 1.7</h1>

<?php

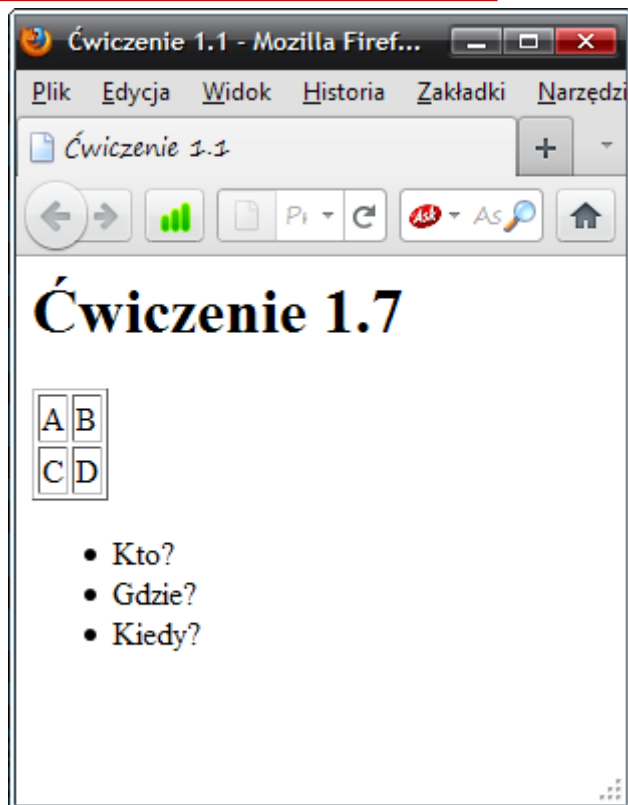
echo '<table border="1">';
echo '<tr>';
echo '<td>A</td>';
echo '<td>B</td>';
echo '</tr>';
echo '<tr>';
echo '<td>C</td>';
echo '<td>D</td>';
echo '</tr>';
echo '</table>';

echo "<ul>";
echo "<li>Kto?</li>";
echo "<li>Gdzie?</li>";
echo "<li>Kiedy?</li>";
echo "</ul>";

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.7:

```
<body>

<h1>Ćwiczenie 1.7</h1>

<table border="1"><tr><td>A</td><td>B</td></tr><tr><td>C ...
</body>
</html>
```

## 1.8 Ćwiczenie 1.8

**TEMAT:** napisy rozciągające się na kilka wierszy.

W języku php , możemy korzystać z napisów zajmujących kilka linii. Rozwiązanie takie będzie bardzo wygodne do umieszczania długich zapytań SQL wewnątrz skryptu php.

```
<body>

<h1>Ćwiczenie 4.8</h1>

<?php

echo "Ślimak, ślimak, wystaw rogi,
dam ci sera na pierogi,
jak nie sera, to kapusty
- od kapusty będziesz tłusty.";

echo '<hr />';
```

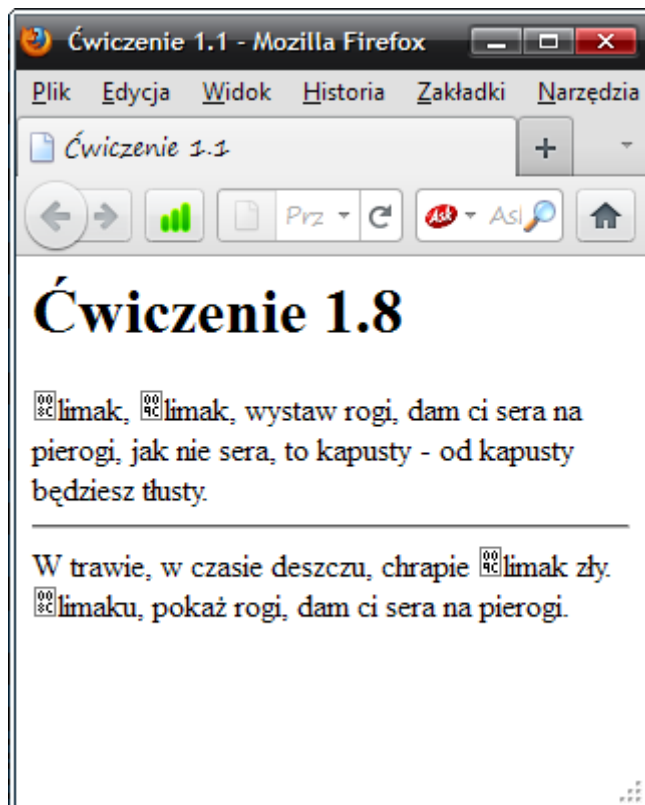


```
echo 'W trawie, w czasie deszczu,  
chrapie ślimak zły.  
Ślimaku, pokaż rogi,  
dam ci sera na pierogi. ';
```

```
?>
```

```
</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.8:

```
<body>
```

```
<h1>Ćwiczenie 1.8</h1>
```

```
Ślimak, ślimak, wystaw rogi,  
dam ci sera na pierogi,  
jak nie sera, to kapusty  
- od kapusty będziesz tłusty.<hr />'W trawie, w czasie deszczu,  
chrapie ślimak zły.  
Ślimaku, pokaż rogi,  
dam ci sera na pierogi. </body>
```

Zauważmy brak białych znaków dookoła znacznika <hr />

## 1.9 Ćwiczenie 1.9

**TEMAT:** Umieszczanie znaku złamania wiersza w kodzie HTML.

Jeśli chcemy, by kod strony WWW (tj. kod HTML) produkowany przez skrypt php był ładnie podzielony na wiersze, to musimy umieścić pomiędzy niektórymi znacznikami HTML znak złamania wiersza. Znakiem tym jest znak `\n` umieszczony wewnątrz napisu otoczonego znakami cudzysłowu, np. `ROBOTA\nPRACA\nHARÓWKA`.

```
<body>

<h1>Ćwiczenie 1.9</h1>

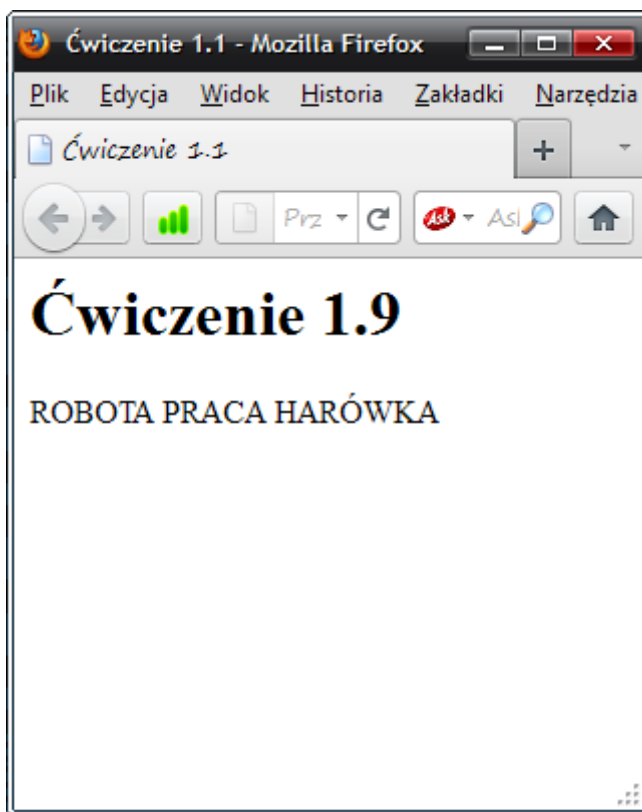
<?php

echo "ROBOTA\nPRACA\n\n\n\nHARÓWKA";

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.9:

```
<body>

<h1>Ćwiczenie 1.9</h1>

ROBOTA
PRACA
```

HARÓWKA  
</body>

Na listingu pojawia się sekwencja \n. Zamiast tego widzimy, że po wyrazach ROBOTA oraz PRACA następuje złamanie wiersza (po wyrazie PRACA czterokrotne). Zwróćmy uwagę na brak złamania wiersza na rysunku. Dzieje się tak dlatego, że znak złamania wiersza (podobnie jak i spacja) pełni rolę separatora wyrazów. Dowolna liczba białych znaków pomiędzy wyrazami oddziela jedynie te wyrazy.

## 1.10 Ćwiczenie 1.10

**TEMAT:** znak \n i napisy otoczone znakami apostrofów.

Jeśli napis jest otoczony znakami apostrofów, to znak \n nie powoduje złamania wiersza!

```
<body>

<h1>Ćwiczenie 1.10</h1>

<?php

echo '<p>\n';
echo 'Oto moja strona...\n';
echo '</p>\n';

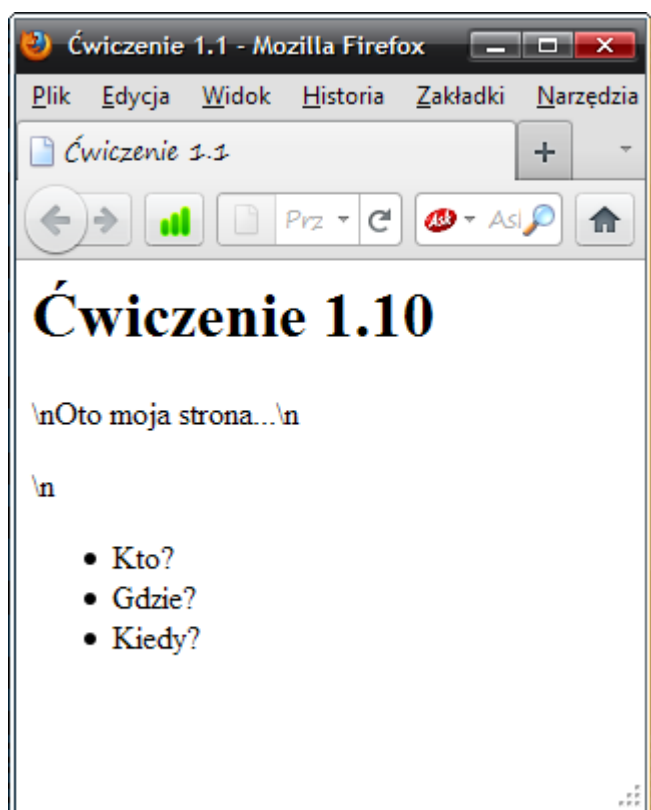
echo "\n\n\n\n\n\n";

echo "<ul>\n";
echo "  <li>Kto?</li>\n";
echo "  <li>Gdzie?</li>\n";
echo "  <li>Kiedy?</li>\n";
echo "</ul>\n";

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.10:

```
<body>

<h1>Ćwiczenie 1.10</h1>

<p>\nOto moja strona...\n</p>\n

<ul>
  <li>Kto?</li>
  <li>Gdzie?</li>
  <li>Kiedy?</li>
</ul>

</body>
```

Jak widać na listingu , znaki \n pojawiają się w kodzie HTML. Nie powoduje to oczywiście pożądanego efektu, a jedynie "zaśmieca" stronę WWW. Na rysunku widać kilka niepotrzebnych napisów \n.

## 11.11 Ćwiczenie 1.11

**TEMAT:** cytowanie znaków specjalnych w napisach otaczanych znakami cudzysłowu.

Znak \ (ang. backslash) usuwa specjalne znaczenie znaku bezpośrednio po nim następującego. Znakami, które mają specjalne znaczenie w napisach otoczonych znakami cudzysłowu są między innymi znaki cudzysłowu ", znak dolara \$ oraz znak backslash \. Dowolny z tych znaków możemy umieścić wewnątrz napisu poprzedzając go backslashem.

```
<body>

<h1>Ćwiczenie 1.11</h1>

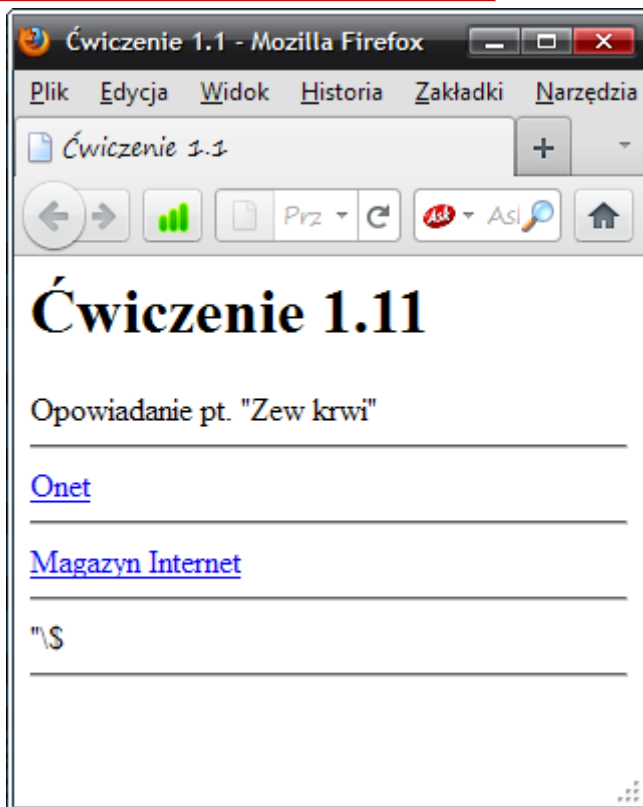
<?php

echo "Opowiadanie pt. \"Zew krwi\"";
echo "\n<hr />\n";
echo "<a href=\"http://www.onet.pl\">Onet</a>";
echo "\n<hr />\n";
echo '<a href="http://www.mi.com.pl">Magazyn Internet</a>';
echo "\n<hr />\n";
echo "\"";    //drukujemy cudzysłów
echo "\\";    //drukujemy backslash
echo "\$";    //drukujemy dolara
echo "\n";    //drukujemy znak złamania wiersza
echo "\n<hr />\n";

?>

</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.11:

```
<body>

<h1>Ćwiczenie 1.11</h1>

Opowiadanie pt. "Zew krwi"
<hr />
<a href="http://www.onet.pl">Onet</a>
<hr />
<a href="http://www.mi.com.pl">Magazyn Internet</a>
<hr />
"&#36;"

<hr />

</body>
```

## 1.12 Ćwiczenie 1.12

**TEMAT:** Drukowanie kodu HTML zawierającego wcięcia.

```
<body>

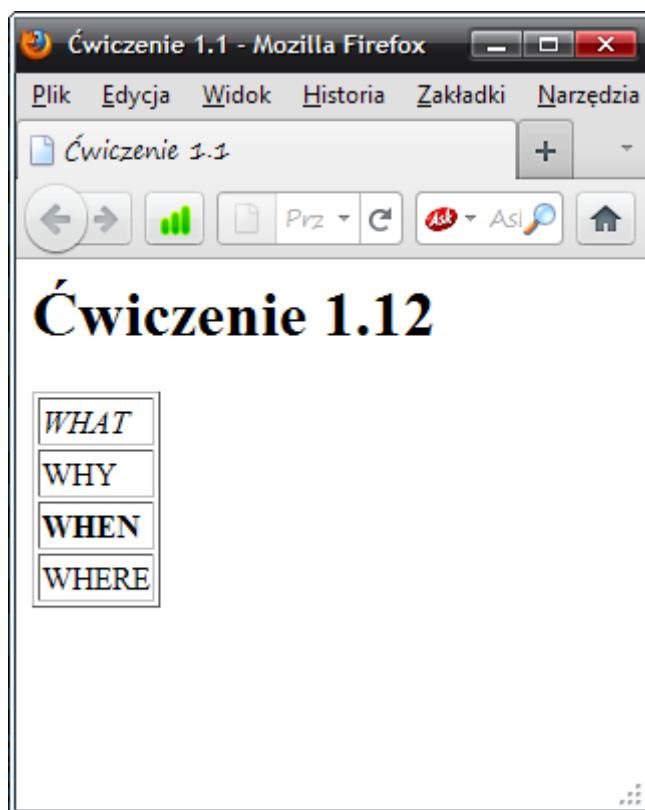
<h1>Ćwiczenie 1.12</h1>

<?php

echo "<table border=\"1\">\n";
echo "  <tr><td><em>WHAT</em></td></tr>\n";
echo "  <tr><td>WHY</td></tr>\n";
```

```
echo "  <tr><td><strong>WHEN</strong></td></tr>\n";  
echo "  <tr><td>WHERE</td></tr>\n";  
echo "</table>\n";  
  
?>  
  
</body>
```

Wygląd skryptu w oknie przeglądarki:



Źródło strony produkowanej przez skrypt z ćwiczenia 1.12:

```
<body>  
  
<h1>Ćwiczenie 1.12</h1>  
  
<table border="1">  
  <tr><td><em>WHAT</em></td></tr>  
  <tr><td>WHY</td></tr>  
  <tr><td><strong>WHEN</strong></td></tr>  
  <tr><td>WHERE</td></tr>  
</table>  
  
</body>
```

### 1.13 Ćwiczenie 1.13

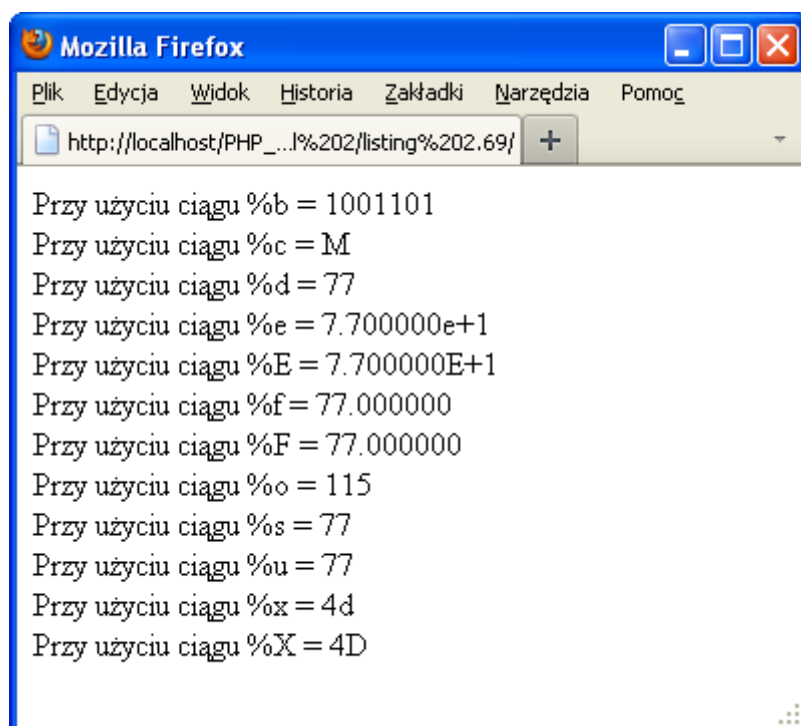
**TEMAT:** Formatowanie ciągów (printf).

```

1 <?php
2 $wartosc = 77;
3 printf("Przy użyciu ciągu %%b = %b<br />", $wartosc);
4 printf("Przy użyciu ciągu %%c = %c<br />", $wartosc);
5 printf("Przy użyciu ciągu %%d = %d<br />", $wartosc);
6 printf("Przy użyciu ciągu %%e = %e<br />", $wartosc);
7 printf("Przy użyciu ciągu %%E = %E<br />", $wartosc);
8 printf("Przy użyciu ciągu %%f = %f<br />", $wartosc);
9 printf("Przy użyciu ciągu %%F = %F<br />", $wartosc);
10 printf("Przy użyciu ciągu %%o = %o<br />", $wartosc);
11 printf("Przy użyciu ciągu %%s = %s<br />", $wartosc);
12 printf("Przy użyciu ciągu %%u = %u<br />", $wartosc);
13 printf("Przy użyciu ciągu %%x = %x<br />", $wartosc);
14 printf("Przy użyciu ciągu %%X = %X<br />", $wartosc);
15 ?>
16

```

Wygląd skryptu w oknie przeglądarki:



## 1.14 Ćwiczenie 1.14

**TEMAT:** Wykorzystanie printf.

Jeśli chcemy dokładnie określić, w jaki sposób różne wartości mają być przedstawione użytkownikowi, możemy skorzystać z funkcji printf. Jej wywołanie ma postać:

Printf(„format”[,argument1[,argument2[,..., argumentN]]])

Format jest ciągiem formatującym i określa, w jaki sposób będą prezentowane argumenty. Ogólna postać tego ciągu to:

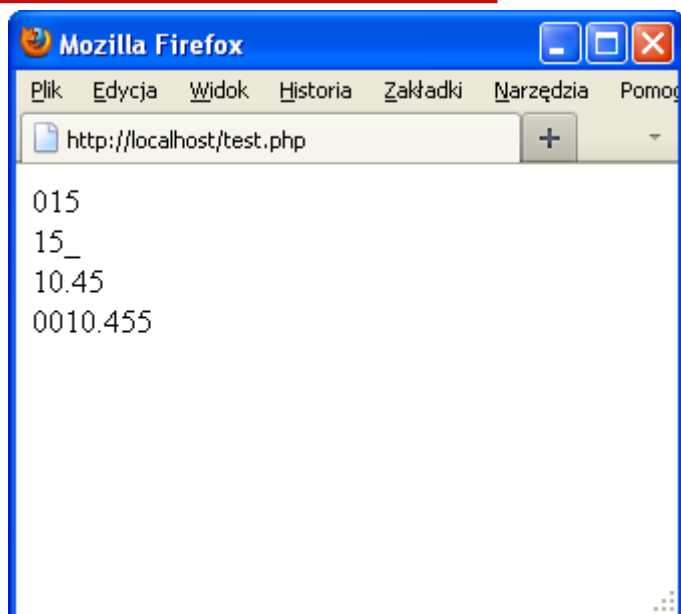
%[z][‘wypełnienie’][-[długość][precyzja]kod

- % - znak rozpoczynający ciąg formatujący.
- z – określenie znaku wartości; można użyć + lub -.
- Wypełnienie – znak, który zostanie użyty jako wypełnienie.
- - - znak określający, że ciąg wynikowy ma być wyrównany do lewej strony (domyślnie jest do prawej).
- Długość – wartość określająca długość ciągu wynikowego.
- Precyzja – wartość określająca ilość miejsc po przecinku.
- Kod – kod określający, w jaki sposób ma być traktowany argument.

```
1 <?php
2
3 printf("%'03d<br/>", 15);
4 printf("%' _-3d<br/>", 15);
5 printf("%.2f<br/>", 10.454647);
6 printf("%'08.3f<br/>", 10.454647);
7
8 ?>
9
```

Wygląd skryptu w oknie przeglądarki:





### 1.15 Ćwiczenie 1.15

**TEMAT:** Wykorzystanie składni heredoc

W przypadku składni heredoc łańcuch znakowy należy rozpocząć od sekwencji <<<, po której musi nastąpić identyfikator. Tego identyfikatora należy następnie użyć w celu zasygnalizowania końca łańcucha znakowego. Dla nazwy identyfikatora obowiązują takie same zasady jak przy nazewnictwie zmiennych. Linia kończąca nie może natomiast zawierać żadnych innych znaków oprócz identyfikatora i średnika.

Wykorzystanie składni heredoc do utworzenia łańcucha znakowego

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
5 <title>Moja strona WWW</title>
6 </head>
7 <body>
8 <div>
9 <?php
10 $napis = <<<ID1
11 Tutaj rozpoczyna się napis.<br />
12 Składa się on z kilku linii.<br />
13 Dlatego też została wykorzystana
14 składnia heredoc.<br />
15 ID1;
16
17 echo $napis;
18 ?>
19 </div>
20 </body>
21 </html>
22
```

## 1.16 Ćwiczenie 1.16

### TEMAT: Łączenie skryptów

Skrypty PHP mogą zawierać dużą ilość kodu i być bardzo skomplikowane. W takiej sytuacji najczęściej kod jest dzielony i zapisywany w kilku oddzielnych plikach, którymi łatwiej zarządzać i analizować. Skoro można pliki dzielić na mniejsze jednostki, musi także istnieć możliwość ich ponownego połączenia. Służy do tego instrukcja ***include***. Napişmy skrypt, którego zadaniem będzie wyświetlanie dowolnej informacji w przeglądarce w taki sposób, że struktura strony HTML będzie zapisana w jednym pliku, natomiast instrukcje echo w drugim.

Treść pliku *skrypt.php*

```
1 <?php
2 echo "<h2 style='text-align:center'>";
3 echo "Witamy na stronie!";
4 echo "</h2>";
5 ?>
6
```

Utworzymy teraz plik, w skład którego będzie wchodził kod HTML realizujący strukturę strony i zawierający dodatkowo instrukcje PHP – ***include***.

Treść pliku *index.php*

```
1 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN">
2 <html>
3 <head>
4 <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-2">
5 <title>Moja strona WWW</title>
6 </head>
7 <body>
8 <?php
9 include "skrypt.php";
10 ?>
11
12 </body>
13 </html>
14
```