

Zajęcia 9

Wprowadzenie do PHP

Co to jest PHP?

PHP jest językiem skryptowym (od wersji 5 w pełni obiektywnym), wykonywanym po stronie serwera. Pozwala on m.in. na tworzenie dynamicznych serwisów WWW (których treść z reguły jest generowana w momencie otwarcia strony serwisu). Język PHP ma jednak więcej zastosowań. Można w nim tworzyć skrypty, wywoływane z wiersza poleceń (podobnie jak skrypty napisane w Perlu, bądź Pythonie). Istnieje również możliwość tworzenia interfejsów graficznych w PHP, z wykorzystaniem biblioteki PHP-GTK.

Jak uruchomić skrypt PHP?

Do uruchomienia skryptu PHP potrzebne są 2 narzędzia:

1. Serwer WWW (np. darmowy Apache)
2. Silnik PHP

Wszystkie potrzebne elementy (również bazę MySQL) można znaleźć w pakiecie XAMPP (<http://www.apachefriends.org/en/xampp.html>).

Wersja LITE (nie wymagająca instalacji) pakietu XAMPP, dostosowana do pracy na komputerach na uczelni jest możliwa do pobrania pod adresem <http://info.wsisiz.edu.pl/~zawadzki/xampp73.zip>.

Kod PHP musi być umieszczony (domyślnie) w plikach z rozszerzeniem .php. Pliki te mogą zawierać zarówno sam kod PHP, jak i kod PHP zagnieżdżony w kodzie HTML (wewnątrz znaczników <?php i ?>).

Przykładowy skrypt PHP

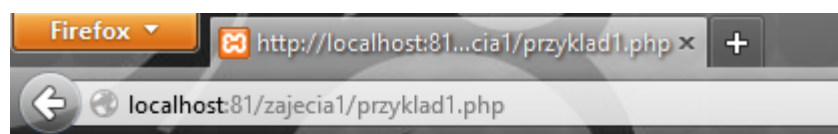
```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
```

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<title>Przykład</title>
<meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <p>Paragraf statyczny</p>

  <?php
    echo "<p>Paragraf dynamiczny</p>";
  ?>
</body>
</html>
```

Powyższy skrypt generuje dwa paragrafy tekstu. Pierwszy z nim jest umieszczony statycznie w kodzie HTML, drugi natomiast jest tworzony poprzez wywołanie instrukcji echo w PHP.

W efekcie wygenerowana strona będzie wyglądała następująco:



Paragraf statyczny

Paragraf dynamiczny

Źródło strony:

```
1 <html>
2   <head>
3     <meta http-equiv="Content-Type" content="text/html; charset=utf-8">
4   </head>
5   <body>
6     <p>Paragraf statyczny</p>
7
8     <p>Paragraf dynamiczny</p>   </body>
9 </html>
```

Jak widać w powyższym przykładzie ważne jest, żeby wszystkie skrypty PHP uruchamiać za pośrednictwem serwera lokalnego (*localhost* w adresie URL), a nie bezpośrednio z dysku komputera, jak miało to miejsce w przypadku statycznych plików HTML.

Składnia

Zmienne

Składnia języka PHP jest bardzo zbliżona do składni innych języków wysokiego poziomu, jak C, C++, czy JAVA.

Zasadniczą różnicą jest podejście do zmiennych, które w PHP są dynamiczne. Oznacza to, że nie ma potrzeby ich deklarowania, jedynie jej nazwa musi być poprzedzona znakiem dolara (\$).

Przykład tworzenia kilku zmiennych:

```
<?php
$zmiennaInt = 51;
$zmiennaFloat = 43.2;
$zmiennaString = "przykładowy tekst";
$tekstZawierajacyZmienne = "integer: $zmiennaInt, float: $zmiennaFloat";
```

W powyższym przykładzie tworzone są zmienne typu integer, float oraz string. Jak widać, w zmiennych tekstowych można zawrzeć nazwę innej zmiennej, co spowoduje wstawienie w danym miejscu wartości zmiennej w trakcie uruchomienia skryptu.

Tablice

Tablice danych w PHP są podstawową strukturą danych. Tablice są budowane dynamicznie, co oznacza, że nie trzeba ich deklarować (długości, typu elementów).

Tablice mogą być również dowolnie indeksowane; można korzystać ze standardowej indeksacji od 0 (albo od dowolnej innej liczby), można też jako indeksy wykorzystać stringi. Tablice w takiej postaci nazywane są tablicami asocjacyjnymi.

```
$tablica = ['red', 'green', 'blue'];
$tablica[] = 'yellow'; // dodawanie elementu na koniec tablicy
$tablica[] = 'orange';
```

```
var_dump($tablica);
```

Zrzut tablicy w przeglądarce:

```
1 array(5) {  
2     [0]=>  
3     string(3) "red"  
4     [1]=>  
5     string(5) "green"  
6     [2]=>  
7     string(4) "blue"  
8     [3]=>  
9     string(6) "yellow"  
10    [4]=>  
11    string(6) "orange"  
12 }  
13
```

Tablice asocjacyjne:

```
$tablicaAsoc = ['klucz1' => 'wartosc1', 'klucz2' => 'wartosc2'];
```

```
$tablicaAsoc['klucz3'] = 'wartosc3';
```

```
var_dump($tablicaAsoc);
```

```
1 array(3) {  
2     ["klucz1"]=>  
3     string(8) "wartosc1"  
4     ["klucz2"]=>  
5     string(8) "wartosc2"  
6     ["klucz3"]=>  
7     string(8) "wartosc3"  
8 }  
9
```

Pętle

W PHP można wykorzystywać pętle o podobnej konstrukcji, jak np. w Javie czy C#. Dostępne są pętle for, while, do...while oraz foreach.

Pętla for

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Przykład 2</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <?php
    for($i = 0; $i < 10; $i++) {
      echo "<p>Paragraf #<?=$i ?></p>";
    }
  ?>

  <?php for($i = 0; $i < 10; $i++): ?>
    <p>Paragraf #<?=$i ?></p>
  <?php endfor; ?>
</body>
</html>
```

Powyższe dwa sposoby wykorzystania pętli for są równoważne (dają ten sam efekt). Przy większej ilości kodu zalecana jest jednak składnia blokowa (druga wersja pętli), gdyż tworzony kod jest dużo bardziej czytelny.

Każda instrukcja blokowa w PHP (pętle, warunki) posiada analogiczne dwie odmiany składni.

W wersji blokowej bardzo często wykorzystywana jest skrócona konstrukcja <?=\$zmienna ?> służąca do wyświetlenia zawartości jednej zmiennej.

Pętla foreach

Pętla *foreach* jest bardzo przydatna do iterowania po tablicach asocjacyjnych, gdyż daje dostęp zarówno do klucz elementu tablicy, jak i wartości:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <title>Przykład 3</title>
  <meta name="viewport" content="width=device-width, initial-scale=1">
</head>
<body>
  <?php
    $jezyki = [
      'Język 1' => 'PHP', 'Język 2' => 'C', 'Język 3' => 'C++',
      'Język 4' => 'Java', 'Język 5' => 'C#', 'Język 6' => 'Ruby',
      'Język 7' => 'Python'
    ];

    foreach($jezyki as $klucz => $wartosc) {
      echo "<p>$klucz: <strong>$wartosc</strong>";
    }
  ?>

  <?php foreach($jezyki as $klucz => $wartosc): ?>
    <p><?=$klucz ?>: <strong><?=$wartosc ?></strong>
  <?php endforeach; ?>
</body>
</html>
```

Zadania

1. Proszę zmodyfikować przykład wykorzystujący pętlę for tak, aby:
 - a. Tekst generował się w tabeli HTML, zamiast w osobnych paragrafach
 - b. Co drugi wiersz tabeli miał ustawiony żółty kolor tła
2. Proszę zmodyfikować przykład wykorzystujący pętlę foreach tak, aby:
 - a. Przetwarzana była tablica zawierająca nazwy kolorów (w formacie HTML)
 - b. Generowana była lista nieuporządkowana (ul/li) z tekstem w kolorze elementu tablicy.