

## IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

## KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

## TWÓJ KOSZYK

DODAJ DO KOSZYKA

## CENNIK I INFORMACJE

ZAMÓW INFORMACJE  
O NOWOŚCIACH

ZAMÓW CENNIK

## CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

# PHP5. Tworzenie stron WWW. Ćwiczenia praktyczne

Autorzy: Andrzej Kierzkowski, Robert Janeczek

ISBN: 83-7361-361-7

Format: B5, stron: 188



Styczne strony WWW już nie spełniają Twoich oczekiwań? Chcesz zamieścić na swojej stronie liczniki, forum dyskusyjne, księgę gości, ankietę i inne elementy interaktywne? Może potrzebny Ci jest mechanizm umożliwiający generację treści strony na podstawie bazy danych? Rozszerz swoją wiedzę o tworzenie dynamicznych stron WWW działających w oparciu o język PHP 5 – popularny, efektywny, działający na wielu platformach i dostępny bezpłatnie język programowania interpretowany po stronie serwera. Został on zaprojektowany specjalnie w celu tworzenia aplikacji WWW i rozwija się szybko. Warto również dodać, że jest bardzo wygodny w użyciu.

Dzięki książce „PHP5. Tworzenie stron WWW. Ćwiczenia praktyczne” poznasz podstawy tworzenia aplikacji WWW w tym języku. Dowiesz się, jak obsługiwać z poziomu PHP formularze, sesje i pliki, nauczysz się łączyć stronę z bazą danych, a przede wszystkim – stworzysz skrypty, dzięki którym Twoja witryna WWW stanie się o wiele bardziej atrakcyjna dla odwiedzających i wygodniejsza w administrowaniu.

- Instalacja i konfiguracja PHP5
- Podstawy języka PHP – zmienne, stałe, operatory i instrukcje
- Pobieranie danych z formularzy
- Obsługa sesji, plików i baz danych
- Tworzenie grafiki z poziomu PHP
- Elementy witryn WWW – forum dyskusyjne, księga gości, ring i inne



# Spis treści

<b>Wstęp</b> .....	<b>5</b>
<b>Rozdział 1. Pierwsze spojrzenie</b> .....	<b>7</b>
1.1. Client-side vs. server-side.....	7
1.2. Witryna PHP.....	11
1.3. Apache, PHP, MySQL — konta w internecie.....	14
1.4. Gotowe skrypty.....	15
<b>Rozdział 2. Instalacja i konfiguracja</b> .....	<b>17</b>
<b>Rozdział 3. Podstawy PHP</b> .....	<b>25</b>
3.1. Pierwsze skrypty.....	26
3.2. Zmienne, stałe, operatory.....	28
3.3. Instrukcja warunkowa.....	34
3.4. Pętla for.....	37
3.5. Pętle while i do... while.....	40
3.6. Instrukcja wyboru (switch).....	43
3.7. Funkcje.....	44
3.8. Złożone struktury danych — tablice zwykłe i tablice asocjacyjne.....	58
<b>Rozdział 4. Dane — pobieranie, przekazywanie i przechowywanie</b> .....	<b>67</b>
4.1. Pobieranie danych z formularzy.....	67
4.2. Cookies.....	78
4.3. Obsługa plików.....	84
4.4. Sesje.....	91
4.5. Baza danych.....	97
<b>Rozdział 5. Grafika w PHP</b> .....	<b>113</b>
<b>Rozdział 6. Trudniejsze zadania webmasterskie</b> .....	<b>129</b>
6.1. Licznik tekstowy.....	129
6.2. Licznik graficzny.....	137
6.3. Księga gości.....	142
6.4. Newsy na stronie.....	146
6.5. Ankieta.....	154
6.6. Ring.....	158
6.7. Galeria zdjęć.....	162
6.8. Analiza dzienników serwera.....	165
6.9. Forum dyskusyjne.....	169

<b>Rozdział 7. Klasy i obiekty</b> .....	<b>175</b>
7.1. Obiektowy licznik.....	175
7.2. Biblioteka PEAR .....	180
<b>Dodatek A Biblioteczka webmastera</b> .....	<b>187</b>

## Rozdział 4.

# Dane — pobieranie, przekazywanie i przechowywanie

Programy PHP, które pisaliśmy do tej pory, nie były bardzo rozbudowane i nie szokowały swoimi możliwościami. Nie stanowiły też rozwiązania typowych problemów webmasterskich. Pewnie już zauważyłeś, że opierały się na komunikacji jednostronnej: program coś wyliczał i wyświetlał wyniki w przeglądarce. Nie można było mu jednak w żaden sposób przekazać danych ani też zapisać na serwerze wyników działania — w celu wykorzystania ich w przyszłości. Nie było też możliwości przekazywania wyników działania z jednej strony do drugiej. W tym rozdziale zajmiemy się tymi właśnie zagadnieniami: pobieraniem danych od użytkownika, pamiętaniem ich zarówno po stronie klienta, jak i serwera oraz przekazywaniem informacji pomiędzy stronami.

### 4.1. Pobieranie danych z formularzy

Podstawowym sposobem pobierania danych od użytkownika na stronach WWW są formularze. Zakładam, że mając styczność z HTML-em, zapoznałeś się już z tym tematem. Warto jednak przypomnieć kilka niezbędnych informacji.

Formularz na stronie WWW ma następującą postać:

```
<FORM ACTION="skrypt.php" METHOD=POST>  
<!--Tu występują pola formularza -->  
<INPUT TYPE=Submit VALUE="Wyślij">  
</FORM>
```

Metodą w formularzu może być także GET (jest to zresztą metoda domyślna: jeżeli nie wpiszesz w definicji formularza żadnej, zostanie użyta właśnie ta). Metody różnią się między sobą sposobem przekazywania danych do serwera. Jeżeli użyjesz metody GET, dane z formularza zostaną dołączone do adresu skryptu odbierającego formularz po znaku zapytania. Adres będzie więc miał postać:

```
http://adres.serwera/skrypt.php?pole1=wart1&pole2=wart2&...&poleN=wartN
```

W przypadku formularzy z dużą liczbą danych taka metoda może być niewygodna. Ma jednak jedną niewątpliwą zaletę: można ją stosować nie tylko do obsługi formularzy, ale po prostu w łańcu, które użytkownik będzie nawet mógł dodać do zakładek.

Metoda POST z kolei przekazuje dane z formularza w inny sposób (po nagłówkach zlecenia HTTP). Danych z formularza nie zobaczymy więc w adresie, ale można ich za to przesłać o wiele więcej.

W formularzu możesz używać: pól tekstowych, pól haseł, pól wyboru, przycisków opcji, obszarów tekstowych, menu, pól wyboru pliku do przesłania. Trudno w tym miejscu szczegółowo omawiać wszelkie elementy formularzy, dlatego zachęcam do zapoznania się z książką *Po prostu HTML 4. Wydanie III*, a w szczególności z jej 16. rozdziałem („Formularze”).

Pobieranie danych z formularza przez skrypt jest trywialnym zadaniem. Dane wpisane do pól formularza są pamiętane w specjalnych tablicach (`$_GET` i `$_POST`), a indeksy, pod którymi są dostępne, noszą takie nazwy, jak pola formularza. Jeżeli formularz zostanie przesłany metodą GET, to wartości wpisane w pola znajdziesz w odpowiednich polach tablicy `$_GET`. Analogicznie jeżeli metodą przesłania formularza będzie POST, to dane będziesz mógł odczytać z tablicy `$_POST`.

#### Ćwiczenie 4.1.

*Utwórz program, który będzie wyświetlał formularz z jednym polem tekstowym, a po wysłaniu tego formularza wyświetli wpisaną wartość.*

W zależności od tego, czy zmienna `$tekst` (tekst będzie nazwą pola w formularzu) przyjmuje jakąś wartość, czy też nie, wyświetlimy albo formularz, albo informację o tym, co w nim zostało wpisane.

**4-01.php**

```
<HTML>
<HEAD>
  <META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-2">
  <TITLE>Formularz</TITLE>
</HEAD>
<BODY>
  <? // Drukuje formularz i jednocześnie odbiera i wyświetla wpisane w nim dane.

  if ($_GET['tekst']) { // jest wpisana jakaś wartość w formularzu
    $tekst = $_GET['tekst'];
    print "wpisana wartość to <B>$tekst</B><BR>";
    print '<A HREF="4-01.php">Powrót do formularza</A>';
  } else { // nie ma wpisanych danych, wyświetlamy formularz
    print '<FORM ACTION="4-01.php" METHOD=GET>';
    print '<INPUT TYPE="text" NAME="tekst">';
```

```
        print '<INPUT TYPE="submit" VALUE="Wyślij">';
        print '</FORM>';
    }

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

---

Warto sprawdzić, jak działa skrypt, zarówno z wykorzystaniem metody GET, jak i POST. Pamiętaj, że poza zmianą metody w formularzu musisz też zmienić nazwę tablicy, z której odczytywane są dane, na `$_POST`. Zauważ różnicę w adresie po wysłaniu formularza za pomocą jednej i drugiej metody.

Wpisz w pole formularza tekst: `<H1>tekst</H1>`. Pojawi się niespodziewany efekt — wyświetlony tekst zostanie wypisany dużą czcionką. Dzieje się tak, ponieważ wpisany tekst wklejamy bezpośrednio do kodu strony. Jeszcze ciekawszy efekt uzyskasz, wpisując tekst: `<script>alert(123);</script>`.

---

Jak widzisz, łatwo stracić kontrolę nad tym, co wyświetla strona, jeżeli wyświetlane na niej dane pochodzą bezpośrednio z formularza. Sprawa nie jest poważna, jeżeli chodzi tylko o stronę wysyłaną pojedynczemu klientowi po wpisaniu przez niego danych. Można sobie jednak wyobrazić księgę gości z wpisami gromadzonymi w bazie danych i wyświetlanymi na żądanie. Jeżeli do bazy będziemy wpisywać nieobrobione dane z formularza, efekty działania odwiedzających (czasem zupełnie niepożądane) będą oglądali wszyscy. Z tego powodu powinieneś stosować funkcję „oczyszczającą” wpisywane dane, szczególnie takie, które mogą się pojawić na stronach WWW.

---

#### Ćwiczenie 4.2.

*Popraw program z ćwiczenia 4.1 tak, by przetwarzał wpisane dane, aby były bezpieczne w wyświetlaniu. Sprawdź jego działanie na przykładach z poprzedniego ćwiczenia.*

W przyszłości zapewne przygotujesz własną funkcję „oczyszczającą” wprowadzane dane. W zależności od ich charakteru, funkcja powinna wykonywać różne operacje. Na początek wykorzystaj wbudowaną funkcję PHP, która znaczniki HTML-a przetworzy na „bezpieczne” do wyświetlenia. W szczególności zamieni znaki większości i mniejszości na `&gt;` i `&lt;`, co spowoduje, że wpisane znaczniki HTML nie zostaną przy ich dołączeniu do strony zinterpretowane, lecz wyświetlone.

**htmlspecialchars**

Zamienia znaczniki HTML na kody „bezpieczne”