

MARCIN LIS

# PHP7



Wszelkie prawa zastrzeżone. Nieautoryzowane rozpowszechnianie całości lub fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonywanie kopii metodą kserograficzną, fotograficzną, a także kopiowanie książki na nośniku filmowym, magnetycznym lub innym powoduje naruszenie praw autorskich niniejszej publikacji.

Wszystkie znaki występujące w tekście są zastrzeżonymi znakami firmowymi bądź towarowymi ich właścicieli.

Autor oraz Wydawnictwo HELION dołożyli wszelkich starań, by zawarte w tej książce informacje były kompletne i rzetelne. Nie biorą jednak żadnej odpowiedzialności ani za ich wykorzystanie, ani za związane z tym ewentualne naruszenie praw patentowych lub autorskich. Autor oraz Wydawnictwo HELION nie ponoszą również żadnej odpowiedzialności za ewentualne szkody wynikłe z wykorzystania informacji zawartych w książce.

Opieka redakcyjna: Ewelina Burska

Projekt okładki: Studio Gravite/Olsztyn

Obarek, Pokoński, Pazdrijowski, Zaprucki

Materiały graficzne na okładce zostały wykorzystane za zgodą Shutterstock.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 32 231 22 19, 32 230 98 63

e-mail: [helion@helion.pl](mailto:helion@helion.pl)

WWW: <http://helion.pl> (księgarnia internetowa, katalog książek)

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/php7pk>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

ISBN: 978-83-283-2478-7

Copyright © Helion 2017

Printed in Poland.

- [Kup książkę](#)
- [Poleć książkę](#)
- [Oceń książkę](#)

- [Księgarnia internetowa](#)
- [Lubię to! » Nasza społeczność](#)

# Spis treści

<b>Wstęp</b> .....	<b>7</b>
O książce .....	7
Narzędzia .....	8
Listingi i rozwiązania ćwiczeń .....	9
<b>Rozdział 1. Podstawy</b> .....	<b>11</b>
Lekcja 1. Czym jest PHP? .....	11
Język skryptowy .....	11
Krótka historia PHP .....	12
Jak to działa? .....	13
Lekcja 2. Instalacja i konfiguracja narzędzi .....	14
Samodzielna instalacja narzędzi .....	14
Korzystanie z pakietu XAMPP .....	30
Plik konfiguracyjny PHP .....	39
PHP w wierszu poleceń .....	40
<b>Rozdział 2. Elementy języka</b> .....	<b>43</b>
Lekcja 3. Pierwszy skrypt .....	43
Zaczynamy .....	43
Znaczniki PHP .....	47
Komentarze .....	49
Wyświetlanie informacji .....	51
Instrukcja print .....	53
Łączenie skryptów .....	53
Lekcja 4. Zmienne, typy danych i operatory .....	59
Czym są zmienne? .....	59
Rodzaje zmiennych, czyli typy danych .....	60
Zmienne w kodzie skryptu .....	65
Wyświetlanie wartości zmiennych .....	66
Operacje na zmiennych .....	71
Operatory .....	71
Zmienne globalne (superglobalne) .....	87
Konwersje typów .....	89
Ćwiczenia do samodzielnego wykonania .....	93
Lekcja 5. Instrukcje sterujące .....	93
Instrukcje warunkowe .....	93
Instrukcja wyboru .....	101
Operator warunkowy .....	103

Pętle .....	104
Składnia alternatywna .....	115
Ćwiczenia do samodzielnego wykonania .....	117
Lekcja 6. Funkcje .....	118
Definiowanie funkcji .....	118
Argumenty funkcji .....	119
Zwracanie wartości przez funkcje .....	121
Zasięg zmiennych .....	123
Sposoby przekazywania argumentów .....	128
Domyślne argumenty funkcji .....	130
Argumenty i typy danych .....	131
Ćwiczenia do samodzielnego wykonania .....	133
Lekcja 7. Obsługa daty i czasu .....	134
Wyświetlanie daty i czasu .....	134
Tworzenie znacznika czasu .....	142
Pozostałe funkcje .....	145
Ćwiczenia do samodzielnego wykonania .....	148
Lekcja 8. Ciągi znaków .....	148
Rodzaje ciągów znaków .....	148
Wyrażenia złożone w ciągach znaków .....	151
Która metoda jest szybsza? .....	152
Formatowanie ciągów .....	153
Przetwarzanie ciągów znaków .....	159
Porównania .....	162
Przeszukiwanie .....	163
Ćwiczenia do samodzielnego wykonania .....	165
Lekcja 9. Tablice .....	165
Proste tablice .....	165
Tablice asocjacyjne .....	169
Operacje na tablicach .....	171
Ćwiczenia do samodzielnego wykonania .....	178
<b>Rozdział 3. Współpraca z systemem plików .....</b>	<b>181</b>
Lekcja 10. Operacje na strukturze systemu plików .....	181
Odczyt zawartości katalogu .....	181
Operacje na katalogach .....	187
Operacje na plikach .....	189
Miejsce na dysku .....	191
Rekurencyjne usuwanie zawartości katalogu .....	192
Nawigacja po katalogach .....	193
Ćwiczenia do samodzielnego wykonania .....	196
Lekcja 11. Operacje na plikach .....	196
Tworzenie i otwieranie plików .....	197
Zamykanie plików .....	199
Odczyt danych .....	199
Zapis danych .....	207
Inne operacje .....	208
Ćwiczenia do samodzielnego wykonania .....	212
Lekcja 12. Praktyczne wykorzystanie plików .....	212
Tekstowy licznik odwiedzin .....	212
Licznik wykorzystujący grafikę .....	215
Lista odnośników .....	216
Lista odwiedzin .....	218
Ćwiczenia do samodzielnego wykonania .....	221

<b>Rozdział 4. Współpraca z przeglądarką .....</b>	<b>223</b>
Lekcja 13. Odbieranie danych z przeglądarki .....	223
Formularze HTML .....	223
Wysyłanie metodą GET .....	224
Metoda POST .....	229
Tablica REQUEST .....	231
Zapis danych w praktyce .....	232
Wysyłanie plików do serwera .....	236
Ćwiczenia do samodzielnego wykonania .....	240
Lekcja 14. Wysyłanie danych do przeglądarki .....	241
Sposoby wysyłania danych .....	241
Wysyłanie zawartości plików .....	242
Sposoby pobierania plików z serwisu .....	245
Ćwiczenia do samodzielnego wykonania .....	259
Lekcja 15. Obsługa cookies .....	259
Krótko o cookies .....	259
Obsługa cookies w PHP .....	260
Korzystanie z cookies .....	264
Ćwiczenia do samodzielnego wykonania .....	269
Lekcja 16. Sesje .....	269
Wstęp do sesji .....	269
Identyfikator sesji .....	270
Rozpoczynanie sesji .....	270
Kończenie sesji .....	271
Zmienne sesji .....	271
Konfiguracja sesji .....	272
Implementacja sesji .....	274
Śledzenie zachowań użytkownika .....	277
Kontrola dostępu z wykorzystaniem sesji .....	279
System logowania z danymi w pliku .....	285
Ćwiczenia do samodzielnego wykonania .....	288
<b>Rozdział 5. Obiektowy PHP .....</b>	<b>289</b>
Lekcja 17. Podstawy obiektowości .....	289
Czym jest obiekt? .....	289
Definicja klasy .....	290
Tworzenie obiektów .....	293
Konstruktory i destruktory .....	295
Automatyczne ładowanie kodu klasy .....	298
Obiektowa lista odwiedzin .....	300
Ćwiczenia do samodzielnego wykonania .....	303
Lekcja 18. Więcej o programowaniu obiektowym .....	303
Dziedziczenie .....	303
Przesłanie składowych .....	307
Klasy i składowe finalne .....	309
Konstruktory i destruktory klas bazowych .....	310
Specyfikatory dostępu .....	312
Składowe statyczne .....	314
Ćwiczenia do samodzielnego wykonania .....	317
Lekcja 19. Wyjątki .....	317
Instrukcja throw .....	318
Klasa Exception i pochodne .....	318
Blok try...catch .....	319

Przechwytywanie wielu wyjątków .....	325
Własne wyjątki .....	327
Ćwiczenia do samodzielnego wykonania .....	328
<b>Rozdział 6. Grafika i obrazy .....</b>	<b>329</b>
Lekcja 20. Obsługa grafiki .....	329
Biblioteka graficzna .....	329
Jak utworzyć galerię obrazów? .....	330
Przetwarzanie grafiki .....	339
Ćwiczenia do samodzielnego wykonania .....	352
<b>Rozdział 7. Obsługa sieci .....</b>	<b>353</b>
Lekcja 21. Połączenia, poczta i FTP .....	353
Tablica \$_SERVER .....	353
Adresy IP .....	356
Jak rozpoznać przeglądarkę? .....	358
Połączenie FTP .....	360
Wysyłanie poczty .....	363
Ćwiczenia do samodzielnego wykonania .....	366
<b>Rozdział 8. Współpraca z bazami danych .....</b>	<b>367</b>
Lekcja 22. Podstawy baz danych .....	367
MySQL i SQLite .....	367
Tabele, klucze i relacje .....	368
Bazy danych a PHP .....	372
Instalacja systemu bazy danych .....	373
Obsługa serwera MySQL .....	380
Lekcja 23. Podstawy SQL .....	386
Czym jest SQL? .....	386
Obsługa tabel .....	387
Typy danych w kolumnach .....	392
Zapytania .....	397
Lekcja 24. PHP i bazy danych .....	410
PHP i MySQL .....	410
PHP i SQLite .....	417
Ćwiczenia do samodzielnego wykonania .....	425
Lekcja 25. Podejście obiektowe .....	426
Korzystanie z PDO .....	426
PHP i SQLite .....	434
Lekcja 26. Bazy danych w praktyce .....	438
Licznik .....	438
Logowanie .....	441
Sonda (ankieta) .....	443
Lista odwiedzin .....	449
Liczba osób na stronie .....	452
Ćwiczenia do samodzielnego wykonania .....	454
<b>Skorowidz .....</b>	<b>455</b>

## Rozdział 2.

# Elementy języka

## Lekcja 3. Pierwszy skrypt

### Zaczynamy

Poznanie nowego języka programowania, niezależnie od tego, czy jest to język skryptowy (tak jak PHP), czy też klasyczny, najlepiej rozpocząć od przykładu, który spowoduje wyświetlenie dowolnego napisu. Napiżemy zatem skrypt, który spowoduje pojawianie się w treści strony HTML pewnego ciągu znaków — niech będzie to napis *Pierwszy skrypt w PHP*. Kod przykładu realizującego to zadanie został przedstawiony na listingu 2.1. Przyjrzyjmy mu się uważnie.

**Listing 2.1.** Skrypt wyświetlający w przeglądarce wybrany tekst (HTML)

---

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Moja strona WWW</title>
  </head>
  <body>
    <p>
      <?php
        echo("Pierwszy skrypt w PHP.");
      ?>
    </p>
  </body>
</html>
```

---

Większa część kodu to standardowe znaczniki HTML tworzące strukturę najprostszej (poprawnej) strony zgodnej z HTML 5. Jak już wspomniano wcześniej, nie będziemy omawiać typowych konstrukcji HTML-a. Przypomnijmy tylko, że struktura taka składa się z deklaracji typu dokumentu (<!DOCTYPE>), nagłówka (<head>) oraz treści (<body>).

W sekcji <head> określany jest sposób kodowania znaków (w tym przypadku jest to standard UTF-8) oraz tytuł strony (znacznik <title>). Ten sam kod przystosowany do standardu XHTML 1.1 w kodowaniu ISO-8859-2 wyglądałby tak, jak przedstawiono to na listingu 2.2.

---

**Listing 2.2.** Skrypt wyświetlający w przeglądarce wybrany tekst (XHTML)

---

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type"
      content="text/html; charset=iso-8859-2" />
    <title>Moja strona WWW</title>
  </head>
  <body>
    <p>
      <?php
        echo("Pierwszy skrypt w PHP.");
      ?>
    </p>
  </body>
</html>
```

---

Różnice są czysto kosmetyczne. Zmieniła się bowiem jedynie deklaracja typu dokumentu oraz treść znacznika <meta> określającego kodowanie znaków. Dodana została również przestrzeń nazw XML. Dla dalszych rozważań nie ma znaczenia, czy użyty zostanie kod HTML, czy XHTML. Podane w książce przykłady będą jednak dostosowane do najbardziej obecnie rozpowszechnionego standardu HTML 5.

W obu przykładach (listingi 2.1 i 2.2) w sekcji <body> wewnątrz znacznika <p> jest umieszczona treść skryptu PHP. Znajduje się ona między znacznikami <?php oraz ?>. Wszystko to, co znajduje się między nimi, stanowi kod PHP i jest przetwarzane przez aparat wykonawczy PHP. W przypadku listingu 2.1 cała treść skryptu to:

```
echo("Pierwszy skrypt w PHP.");
```

Jest to instrukcja, czyli coś, co możemy traktować jako polecenie do wykonania. To polecenie oznacza: wyślij do przeglądarki<sup>1</sup> ciąg znaków znajdujący się między znakami cudzysłowu prostego<sup>2</sup>. Zwróćmy uwagę na występujący na końcu znak średnika. Oznacza on koniec instrukcji<sup>3</sup>.

---

<sup>1</sup> Jest to pewne uproszczenie. W rzeczywistości instrukcja echo powoduje wysłanie danych do standardowego wyjścia, co akurat w omawianym przypadku będzie równoznaczne z wysłaniem tekstu do przeglądarki (za pośrednictwem serwera WWW). Gdyby skrypt został uruchomiony na konsoli, tekst pojawiłby się właśnie na niej.

<sup>2</sup> W rzeczywistości nie są to prawdziwe znaki cudzysłowu, ale tzw. znaki zastępcze cudzysłowu ASCII. Najczęściej jednak mówi się na nie po prostu „cudzysłów” lub „cudzysłów prosty”.

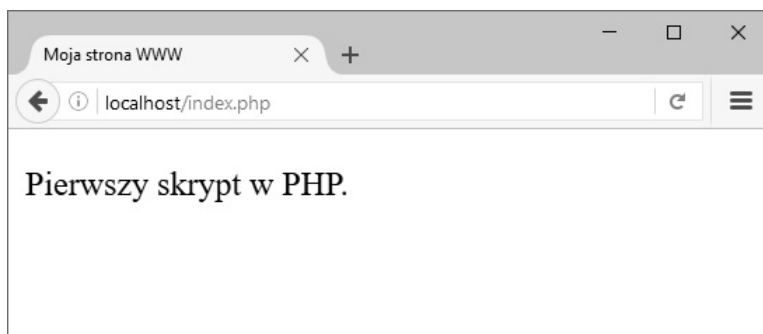
<sup>3</sup> Formalnie rzecz ujmując, średnika trzeba używać do rozdzielania instrukcji lub wtedy, gdy nie kończymy w danym pliku trybu PHP. W tym przypadku średnik można by więc pominąć (mamy do czynienia tylko z jedną instrukcją PHP). Dla zachowania spójności kodu lepiej jednak zawsze dodawać średnik.



Zobaczmy teraz, jak zadziała taki skrypt po wczytaniu do przeglądarki. Całą treść listingu 2.1 zapisujemy w pliku o nazwie *index.php*, po czym umieszczamy go w katalogu głównym serwera WWW (patrz lekcja 2. z rozdziału 1.). Odwołujemy się następnie do pliku *index.php*, wpisując na pasku adresu: *http://localhost/index.php* lub *http://127.0.0.1/index.php*. Zobaczymy widok taki jak na rysunku 2.1, a zatem skrypt działa zgodnie z założeniami.

**Rysunek 2.1.**

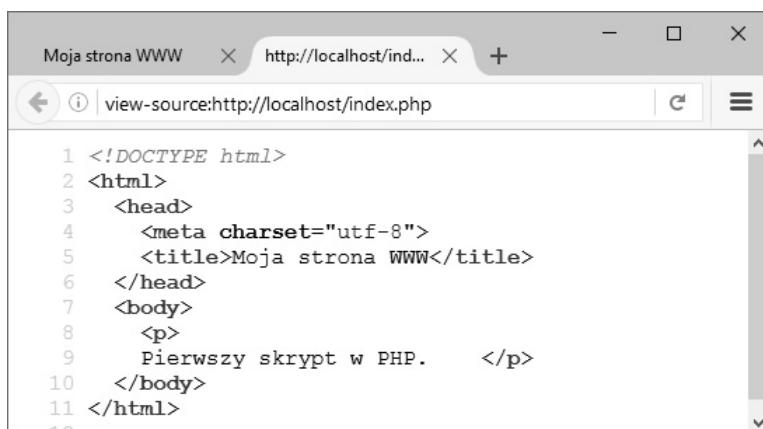
*Efekt działania skryptu z listingu 2.1*



Obejrzymy dodatkowo, jak wygląda kod źródłowy tak wygenerowanej strony. W większości przeglądarek należy w tym celu wybrać odpowiednią pozycję z menu bądź wciśnąć właściwą kombinację klawiszy. W Firefoxie 47 będą to kolejne menu: *Narzędzia, Dla twórców witryn, Źródło strony* bądź skrót klawiaturowy *Ctrl+U*. Sposób wyświetlenia kodu źródłowego strony również zależy od rodzaju przeglądarki, np. w przypadku Firefoksa postać ta będzie podobna do przedstawionej na rysunku 2.2, a sam tekst zawsze będzie miał postać widoczną na listingu 2.3.

**Rysunek 2.2.**

*Kod źródłowy strony w przeglądarce Mozilla Firefox*

**Listing 2.3.** *Kod HTML strony generowanej przez skrypt 2.1*

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Moja strona WWW</title>
  </head>
```

```

<body>
  <p>
    Pierwszy skrypt w PHP.  </p>
</body>
</html>

```

Widać wyraźnie, że zniknęły wszelkie znaczniki PHP i pozostał jedynie czysty kod HTML. Taka jest bowiem istota działania języków skryptowych pracujących po stronie serwera. Zadaniem skryptu PHP jest wygenerowanie takiego kodu, który będzie mógł być zrozumiały dla przeglądarki. W tym przykładzie w miejsce skryptu zawartego między znacznikami `<?php i ?>` została wstawiona wytworzona przez niego treść, czyli napis `Pierwszy skrypt w PHP.` A zatem, powtarzając raz jeszcze: skrypt PHP musi wygenerować taką treść, która zostanie zrozumiana przez przeglądarkę. Najczęściej jest to kod HTML, ale może to być również dokument PDF, plik skompresowany, obraz czy dowolne inne dane. Ważne jest to, żeby dane te zostały wysłane do przeglądarki w taki sposób, aby mogła ona je poprawnie zinterpretować.

Zwróćmy w tym miejscu uwagę, że znacznik kończący akapit (`</p>`) znalazł się w tej samej linii co napis wygenerowany przez skrypt. Dlaczego tak się stało? Otóż wynika to ze sposobu interpretacji kodu przez PHP. Po prostu jako skrypt traktowany jest cały fragment:

```

<?php
  echo("Pierwszy skrypt w PHP.");
?>

```

włącznie ze znajdującym się za znacznikiem `?>` znakiem końca linii. Cały ten fragment jest zamieniany na treść wygenerowaną przez skrypt. Skoro tą treścią jest napis `Pierwszy skrypt w PHP.`, który nie zawiera znaku końca linii, łączy się on z następującym po `?>` znacznikiem HTML. Tym znacznikiem jest `</p>`, a zatem powstaje połączona linia:

```

Pierwszy skrypt w PHP.  </p>

```

Nie ma to znaczenia dla przeglądarki, która w trakcie interpretacji kodu i tak pomija znaki końca linii, ale ma znaczenie dla osoby przeglądającej czy analizującej kod, bo w takiej postaci jest on mniej czytelny. Jak więc zapobiec temu zjawisku? Logika podpowiada, że trzeba po prostu wprowadzić dodatkowy znak końca linii. Można to zrobić na dwa sposoby — albo w instrukcji `echo`, albo w kodzie HTML. Pierwsze rozwiązanie będzie wymagało zastosowania sekwencji znaków specjalnych. Ta tematyka zostanie przedstawiona w kolejnej lekcji. Na razie można powiedzieć tyle, że instrukcja `echo` wyprowadzająca dodatkowy znak końca linii powinna mieć postać:

```

echo("Pierwszy skrypt w PHP.\n");

```

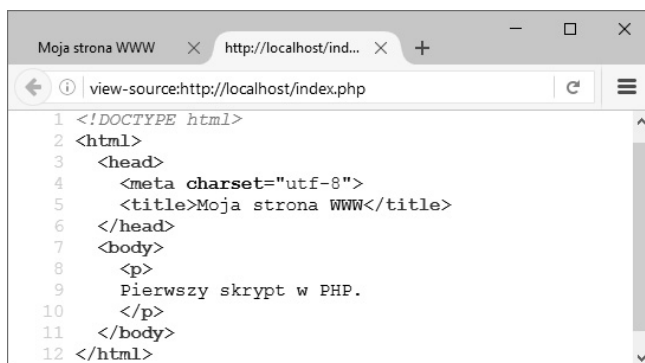
Drugi sposób to wprowadzenie do kodu dodatkowej pustej linii. Cały skrypt miałby w takiej sytuacji postać widoczną na listingu 2.4. Kod źródłowy wygenerowanej przez niego strony (w interpretacji przeglądarki Firefox) jest natomiast widoczny na rysunku 2.3. Tym razem do czytelności listingu nie powinniśmy mieć już żadnych zastrzeżeń.

**Listing 2.4.** Skrypt po wprowadzeniu do kodu dodatkowej pustej linii

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Moja strona WWW</title>
  </head>
  <body>
    <p>
      <?php
        echo("Pierwszy skrypt w PHP.");
      ?>
    </p>
  </body>
</html>

```

**Rysunek 2.3.**  
Kod źródłowy  
strony generowanej  
przez listing 2.4

## Znaczniki PHP

Wiemy już, że treść skryptu PHP musi być oddzielona od kodu HTML specjalnymi znacznikami. Dokładniej rzecz ujmując, każdy fragment kodu PHP (nawet jeśli nie ma wokół niego kodu HTML) musi być ujęty w znaczniki, tak aby aparat wykonawczy PHP wiedział, że dana sekcja to instrukcje PHP, a nie inne dane. Podstawowe znaczniki poznaliśmy w poprzednim punkcie, tak naprawdę jest ich jednak nieco więcej. W sumie można wyróżnić cztery różne typy:

- ♦ znaczniki kanoniczne,
- ♦ znaczniki typu SGML,
- ♦ znaczniki typu ASP,
- ♦ znaczniki skryptów HTML.

Przy czym w PHP7 występują tylko dwa pierwsze typy znaczników (kanoniczne i SGML), natomiast pozostałe były dostępne w poprzednich wersjach PHP.

## Znaczniki kanoniczne

Znaczniki kanoniczne to standardowe znaczniki PHP, które poznaliśmy już wcześniej. Znacznik otwierający to `<?php`, natomiast zamykający to `?>`. Schematyczna konstrukcja wykorzystująca ten typ ma zatem następującą postać:

```
<?php
// Tutaj kod skryptu
?>
```

Znaczniki kanoniczne są rozpoznawane zawsze, niezależnie od tego, jakie opcje są włączone w pliku konfiguracyjnym PHP (*php.ini*). Używanie tych znaczników to również zalecany sposób umieszczania skryptów w kodzie HTML i właśnie ten sposób będzie wykorzystywany w przykładach w dalszej części książki.

## Znaczniki typu SGML

Znaczniki typu SGML to znaczniki w postaci skróconej. Znacznik otwierający to `<?`, zamykający — `?>`. Schematyczna konstrukcja wykorzystująca ten typ będzie miała postać:

```
<?
// Tutaj kod skryptu
?>
```

Jest to najkrótsza forma znaczników bloku PHP, jakie można zastosować. Aby jednak móc korzystać z tego sposobu, należy włączyć rozpoznawanie tychże znaczników. Można to zrobić, wykorzystując jeden z przedstawionych niżej sposobów:

- ◆ Można włączyć opcję `enable-short-tags` przed kompilacją pakietu (`./configure --enable-short-tags`).
- ◆ W pliku konfiguracyjnym *php.ini* można umieścić linię `short_open_tag = 0n`.

Wykorzystywanie tego typu znaczników, choć możliwe, nie zawsze jest jednak zalecane.

## Znaczniki typu ASP

Znaczniki typu ASP są znane użytkownikom technologii ASP. Znacznik otwierający to `<%`, a zamykający — `%>`. Schematyczna konstrukcja wykorzystująca ten typ będzie miała więc postać:

```
<%
// Tutaj kod skryptu
%>
```

Aby móc korzystać z tego typu wyróżnienia bloków PHP, należy w pliku konfiguracyjnym włączyć opcję `asp_tags = 0n`. Możliwość korzystania z tego typu znaczników została usunięta w PHP7.

## Znaczniki skryptów HTML

Ta postać jest dobrze znana użytkownikom HTML. Jest to typowy znacznik `<script>` z parametrem `language` ustawionym na wartość `php`. Znacznik otwierający będzie miał zatem postać `<script language="php">`, natomiast zamykający — `</script>`. W związku z tym schematyczna konstrukcja wykorzystująca ten typ znaczników to:

```
<script language="php">
    // Tutaj kod skryptu
</script>
```

Postać ta, podobnie jak kanoniczna, była rozpoznawana standardowo i nie wymagała włączania żadnych dodatkowych opcji konfiguracyjnych. Ta forma znaczników została wyłączona w PHP7.

## Komentarze

W kodzie PHP można, tak jak w większości innych, zarówno skryptowych, jak i klasycznych języków programowania, stosować komentarze. Są one ignorowane i usuwane w trakcie przetwarzania skryptu, pozwalają za to umieścić w kodzie uwagi, które mogą być dla nas przydatne podczas jego analizy. W pliku wynikowym, który powędruje do użytkownika (przeglądarki), na pewno ich nie będzie.

Do wyboru mamy trzy rodzaje komentarzy: dwa zapożyczone są z języków takich jak C i C++, a jeden — z powłok (ang. *system shell*) uniksowych. Są to:

- ♦ komentarz blokowy,
- ♦ komentarz jednowierszowy,
- ♦ komentarz jednowierszowy uniksowy.

### Komentarz blokowy

Ten typ komentarza zaczyna się od sekwencji znaków `/*`, a kończy się sekwencją `*/`. Wszystko to, co znajduje się między tymi znakami, zostanie zignorowane przez PHP. Przykład poprawnego zastosowania może być taki:

```
<?php
/*W tym miejscu wyświetlamy napis powitalny*/
echo ("Witamy na stronie");
/*Koniec kodu PHP*/
?>
```

Po przetworzeniu takiego kodu przez PHP do przeglądarki zostanie wysłana jedynie linia zawierająca ciąg znaków:

```
Witamy na stronie
```

Należy pamiętać, że komentarze tego typu koniecznie muszą znaleźć się w bloku PHP (co wydaje się zrozumiałe, jako że dotyczy on kodu PHP) oraz że nie wolno ich zagnieźdzać, czyli w jednym komentarzu blokowym nie wolno umieszczać kolejnego. Zabroniona jest zatem taka sytuacja, jak przedstawiona poniżej:

```

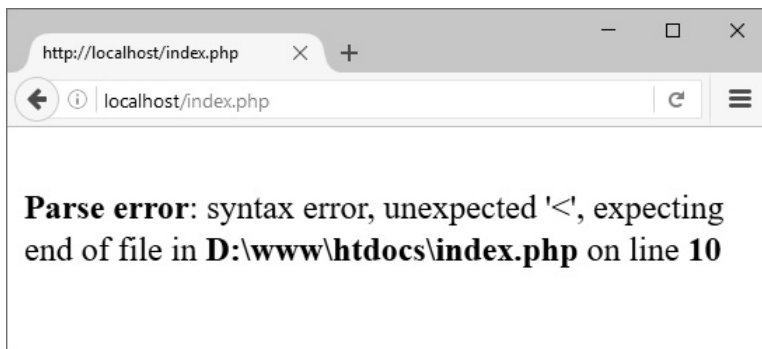
<?php
/*Początek komentarza 1.
   echo("Witamy na stronie");
   /*Komentarz 2.*/
Koniec komentarza 1.*/
?>

```

Próba zastosowania takiej konstrukcji spowoduje powstanie błędu, który objawi się komunikatem przedstawionym na rysunku 2.4<sup>4</sup>. Wewnątrz komentarza blokowego może natomiast znaleźć się komentarz jednowierszowy lub uniksowy, co zostanie pokazane na kolejnych stronach.

### Rysunek 2.4.

Próba zagnieżdżenia komentarzy blokowych powoduje wyświetlenie komunikatu o błędzie



## Komentarz jednowierszowy

Pierwszy typ komentarza jednowierszowego został zaczerpnięty ze składni klasycznych języków programowania takich jak C i C++. Rozpoczyna się on od sekwencji znaków `//` i kończy w bieżącej linii skryptu. Wszystko to, co znajduje się między tymi znakami a końcem linii, jest traktowane jako komentarz i ignorowane przez analizator składniowy. Przykład wykorzystania tego typu komentarza wygląda następująco:

```

<?php
// W tym miejscu wyświetlamy napis powitalny
echo ("Witamy na stronie.");
// Koniec kodu PHP
?>

```

Jak wspomniano już wcześniej, komentarze jednowierszowe mogą być zagnieżdżane wewnątrz komentarza blokowego, np.:

```

<?php
/*Początek komentarza blokowego
// Komentarz jednowierszowy
Koniec komentarza blokowego*/
?>

```

Taka konstrukcja formalnie jest w pełni dopuszczalna i nie spowoduje powstania błędu, jednak w praktyce zazwyczaj nie ma potrzeby jej stosowania.

<sup>4</sup> O ile w opcjach konfiguracyjnych PHP nie zostało wyłączone wyświetlanie komunikatów o błędach.

# Skorowidz

## A

adresy IP, 356  
ankieta, 4 43  
argumenty, 131  
  domyślne, 130  
  funkcji, 119  
  konstruktorów, 297  
atrybut, 290  
  AUTO\_INCREMENT, 388  
  INDEX, 388  
  NOT NULL, 388  
  PRIMARY KEY, 388  
  UNIQUE, 388  
atrybuty formularza, 224  
automatyczne ładowanie kodu, 298

## B

baza danych, 367, 372  
bezpieczeństwo pakietu XAMPP, 34  
biblioteka graficzna, 329  
bitowa różnica symetryczna, 75  
blok try...catch, 319  
brak połączenia, 27

## C

ciasteczka, *Patrz* cookies  
ciągi znaków, 148  
  formatowanie, 153  
  porównania, 162  
  przeszukiwanie, 163  
  przetwarzanie, 159  
  wielkość liter, 158  
  wyrażenia złożone, 151  
  wyświetlanie, 152  
cookies, 259

argumenty, 260  
usuwanie, 263  
zapamiętywanie danych, 264  
zliczanie liczby odwiedzin, 267  
czas, 134  
  generowania strony, 145

## D

data, 134  
DCL, Data Control Language, 387  
DDL, Data Definition Language, 386  
destruktory, 295, 298  
  klas bazowych, 310  
DML, Data Manipulation Language, 387  
dostęp  
  do cookies, 263  
  do plików, 210  
  do składowych, 292, 312  
  do strony, 279  
DSN, Data Source Name, 427  
dziedziczenie, 303

## E

ekran tytułowy pakietu XAMPP, 34  
elementy  
  składowe formularza, 224  
  typu radio, 227

## F

filtry funkcji imagefilter, 348  
format  
  GIF, 340  
  JPEG, 340  
  MIME, 242  
  PNG, 340

formatowanie ciągów, 153  
 formularze, 223  
   atrybuty, 224  
   elementy składowe, 224  
   odczyt danych, 226, 230  
   wprowadzanie opinii, 232  
   wysyłanie plików, 237  
 FTP, 360  
 funkcja  
   chdir, 188  
   checkdate, 146  
   checkPass, 441  
   date, 134, 135, 136  
   disk\_free\_space, 191  
   fclose, 199  
   fgetc, 202  
   fgets, 199  
   file, 206  
   file\_exists, 188, 189  
   file\_put\_contents, 207  
   filesize, 189, 204  
   fopen, 189, 197, 198  
   fread, 203, 204  
   fwrite, 207  
   getcwd, 188  
   getdate, 136, 137  
   getimagesize, 343  
   gmdate, 146  
   imagefilter, 348  
   localtime, 147  
   mkdir, 187  
   mysqli\_affected\_rows, 416  
   mysqli\_close, 412  
   mysqli\_connect, 410  
   printf, 155  
   readData, 451  
   readdir, 192  
   rmdir, 188  
   setlocale, 141  
   sqlite\_array\_query, 423  
   sqlite\_close, 418, 435  
   sqlite\_exec, 422  
   sqlite\_open, 418  
   sqlite\_query, 420  
   sqlite\_unbuffered\_query, 420  
   strftime, 139, 140  
   time, 148  
   unlink, 189  
 funkcje  
   argumenty, 119  
   argumenty domyślne, 130  
   definiowanie, 118  
   przekazywanie argumentów, 128  
   zmieniające wielkość liter, 158

zwracanie wartości, 121

## G

galeria  
   obrazów, 330  
   z miniaturami obrazów, 334  
 generowanie  
   grafiki, 343  
   listy plików, 251  
 głosowanie, 443  
 grafika, 329

## H

hasło, 281  
 HTTP, HyperText Transfer Protocol, 13

## I

identyfikator sesji, 270  
 iloczyn  
   bitowy, 74  
   logiczny, 77  
 implementacja sesji, 274  
 informacja  
   o błędzie, 246  
   o obrazie, 342  
   o pliku, 190  
 instalacja  
   MySQL, 374  
   narzędzi, 14  
   pakietu XAMPP, 31, 35  
   PHP, 16, 21, 23  
   serwera WWW, 15, 20, 23  
   SQLite, 373  
   systemu bazy danych, 373  
   w trybie tekstowym, 20  
 instalator pakietu XAMPP, 32  
 instrukcja  
   break, 109  
   continue, 111  
   CREATE TABLE, 387  
   DELETE, 409  
   DROP TABLE, 392  
   echo, 66  
   if, 94, 115  
   if...else, 95, 115  
   if...else if, 96, 116  
   include, 53  
   INSERT INTO, 397  
   print, 53  
   require, 56



SELECT, 399  
throw, 318  
UPDATE, 407  
instrukcje  
sterujące, 93  
warunkowe, 93, 115  
wyboru, 101

**J**

język  
DCL, 387  
DDL, 386  
DML, 387  
języki skryptowe, 11

**K**

katalogi  
bieżące, 188  
nawigacja, 193  
odczyt zawartości, 181  
rekurencyjne usuwanie zawartości, 192  
tworzenie, 187  
usuwanie, 188  
klasa, 290  
Exception, 318  
PDO, 427  
klasy  
bazowe, 310  
potomne, 310  
klucze, 368  
kod  
formularza, 225  
HTML, 45  
źródłowy strony, 45  
kodowanie znaków specjalnych, 69  
kolory, 340  
komentarz, 49  
blokowy, 49  
jednowierszowy, 50  
jednowierszowy uniksowy, 51  
komunikacja z serwerem, 383  
konfiguracja  
narzędzi, 14  
sesji, 272  
konstruktory, 295  
klas bazowych, 310  
konwersje typów, 89  
kończenie  
połączenia, 411, 418, 428, 435  
sesji, 271

**L**

liczba osób na stronie, 452  
licznik  
odwiedzin strony, 212, 438  
wykorzystujący grafikę, 215  
Linux  
instalacja MySQL, 379  
instalacja pakietu XAMPP, 31  
testowanie instalacji XAMPP-a, 34  
uruchamianie serwera MySQL, 381  
uruchamianie serwerów, 32  
lista  
odnośników, 216, 246  
odwiedzin, 218, 449  
plików, 251  
z identyfikatorami, 254  
logiczna alternatywa wykluczająca, 78  
logowanie, 281, 285, 441  
lokalizacja dołączanych plików, 58

**Ł**

ładowanie kodu klasy, 298  
łączenie skryptów, 53

**M**

Menedżer oprogramowania, 15  
metoda, 290  
exec, 437  
execute, 433  
fetch, 429  
fetchAll, 432  
GET, 224  
POST, 229  
query, 428  
MIME, Multipurpose Internet Mail Extensions, 242  
miniatury obrazów, 334  
Mint  
instalacja PHP, 16  
instalacja serwera WWW, 15  
testowanie instalacji, 18  
uruchamianie serwera, 16  
modyfikacja  
danych, 407  
tabel, 390  
modyfikatory dostępu, 312  
MySQL, 367, 374, 410  
logowanie, 441  
operatory logiczne, 404  
operatory relacyjne, 403, 404

tworzenie konta użytkownika, 385  
uruchamianie serwera, 380

## N

nakładanie filtrów, 348  
narzędzia, 8  
nawiązywanie połączenia, 410, 417, 427  
z bazą, 434  
nawigacja po katalogach, 193  
nazewnictwo zmiennych iteracyjnych, 113  
negacja  
bitowa, 75  
logiczna, 78

## O

obiekty, 289  
obliczanie liczby dni, 143  
obracanie, 351  
obrazy, 329  
obsługa  
cookies, 259  
daty i czasu, 134  
grafiki, 329  
serwera MySQL, 380  
sieci, 353  
tabel, 387  
odbieranie plików przeglądarki, 243  
odeczyt  
bloków danych, 203  
całej zawartości pliku, 204  
danych, 199  
danych z formularza, 226, 230  
danych z tabeli, 430  
pojedynczych wierszy, 199  
pojedynczych znaków, 202  
zawartości katalogu, 181  
zawartości tabeli, 414, 420  
odnośniki, 216  
odwołania do składowych, 293  
opcje konfiguracyjne sesji, 272  
operacje  
na katalogach, 187  
na plikach, 189, 196  
na tablicach, 171  
na zmiennych, 71  
operator, 71  
indeksowania tablic, 83  
kontroli błędów, 84  
kontroli typów, 85  
łańcuchowy, 83  
łączenia tablic, 82

new, 86  
rozdzielania wyrażień, 86  
rzutowania typów, 86  
warunkowy, 84, 103  
wykonania polecenia zewnętrznego, 85

operatory  
arytmetyczne, 71  
bitowe, 73  
inkrementacji i dekrementacji, 80  
logiczne, 77, 404  
porównywania tablic, 83  
przypisania, 78  
relacyjne, 79, 403, 404  
tablicowe, 82  
otwieranie plików, 197

## P

pakiet XAMPP, 30, 380  
panel kontrolny XAMPP-a, 38  
PDO, PHP Data Objects, 426  
pętla  
do...while, 107  
for, 104, 112, 116  
foreach, 107, 116  
while, 106, 117  
PHP, 11  
pierwszy skrypt, 43  
plik  
ankieta.php, 445  
apache2.conf, 29  
glowna.php, 441  
httpd.conf, 29  
index.php, 441  
konfiguracyjny PHP, 29, 39  
logout.php, 441  
php.ini, 39  
pliki  
graficzne, 340  
określanie rozmiaru, 189  
otwieranie, 197  
synchronizacja dostępu, 210  
tworzenie, 189, 197  
usuwanie, 189  
zamykanie, 199  
zapis danych, 207  
zmiana pozycji, 208  
pobieranie  
danych, 399  
plików, 250  
plików z serwera FTP, 361  
plików z serwisu, 245  
wszystkich wierszy tabeli, 400  
wyników zapytania, 432

- zawartości kolumn, 401
- poczta, 363
- podejście obiektowe, 426
- pola, 290
- polecenie ALTER TABLE, 390
- połączenia
  - FTP, 360
  - stałe, 411
  - z bazą danych, 410, 417, 428, 434
  - z serwerem, 27
- porównania ciągów, 162
- priorytety operatorów, 86
- problem
  - błąd podczas uruchamiania, 28
  - brak połączenia, 27
- programowanie obiektowe, 303
- protokół
  - HTTP, 13
  - SMTP, 363
- przechwytywanie
  - wielu wyjątków, 325
  - wyjątku, 320
- przeglądarka, 223
  - odbieranie danych, 223
  - wysyłanie danych, 241
- przekazywanie argumentów, 128
- przesłanie składowych, 307
- przesunięcie bitowe
  - w lewo, 76
  - w prawo, 76
- przeszukiwanie ciągów, 163
- przetwarzanie
  - ciągów znaków, 159
  - grafiki, 339
  - obrazów, 348
- pula, pool, 411

## R

- rekurencyjne usuwanie zawartości katalogu, 192
- relacje, 369
- relacyjne bazy danych, 367
- rodzaje
  - ciągów znaków, 148
  - zmiennych, 60
- rozmiar pliku, 189
- rozpoczynanie sesji, 270
- rozpoznawanie
  - systemu operacyjnego, 360
  - przeglądarki, 358
- rysowanie
  - elips, 346
  - linii, 343
  - prostokątów, 344

- wielokątów, 345
- wycinków elips, 347

## S

- selektywne pobieranie danych, 403
- serwer
  - Apache, 15
  - Apache 2, 23
  - MySQL, 380
  - WWW, 15
- sesje, 269
  - identyfikator, 270
  - implementacja, 274
  - kontrola dostępu, 279
  - kończenie, 271
  - opcje konfiguracyjne, 272
  - rozpoczynanie, 270
  - śledzenie zachowań użytkownika, 277
  - zmienne, 271
- skalowanie, 349
- składnia
  - alternatywna, 115
  - heredoc, 63
  - nowdoc, 64
- składowe
  - finalne, 309
  - klasy, 290
  - statyczne, 314
- skrypt
  - odbierający pliki, 239
  - wysyłający pliki, 247
- słowo kluczowe
  - class, 290
  - new, 293
- SMTP, 363
- sonda, 443
- sortowanie, 172, 184, 400
- specyfikatory dostępu, 312
- SQL, Structured Query Language, 386
- SQLite, 367, 373, 417, 434
- stała
  - FILE\_APPEND, 208
  - FILE\_IGNORE\_NEW\_LINES, 206
  - FILE\_SKIP\_EMPTY\_LINES, 206
  - FILE\_USE\_INCLUDE\_PATH, 206, 208
  - LOCK\_EX, 208, 211
  - LOCK\_SH, 211
  - LOCK\_UN, 211
  - SCANDIR\_SORT\_ASCENDING, 184
  - SCANDIR\_SORT\_DESCENDING, 184
  - SCANDIR\_SORT\_NONE, 184
  - SEEK\_CUR, 209
  - SEEK\_END, 209

SEEK\_SET, 209  
 suma  
   bitowa, 74  
   logiczna, 77  
 system  
   logowania, 285  
   plików, 181

## Ś

śledzenie zachowań użytkownika, 277

## T

tabele, 368  
   modyfikacja, 390  
   tworzenie, 387  
   usuwanie, 392  
 tablica  
   \$\_FILES, 238  
   \$\_SERVER, 353  
   REQUEST, 231  
 tablice  
   asocjacyjne, 169  
   proste, 165  
   sortowanie, 172  
   ustalanie liczby elementów, 172  
   wyświetlanie zawartości, 171  
 tekstowy licznik odwiedzin, 212  
 test działania PHP, 26  
 testowanie  
   instalacji, 18, 22, 26  
   konfiguracji, 38  
 tworzenie  
   ciągów znaków, 149  
   galerii obrazów, 330  
   katalogów, 187  
   konstruktorów, 295  
   konta użytkownika, 385  
   obiektów, 86, 293  
   obrazu, 339  
   plików, 189, 197  
   tabel, 387  
   znacznika czasu, 142  
 typ danych, 60, 131  
   BINARY, 396  
   BLOB, 396  
   boolean, 60, 92  
   CHAR, 395  
   ENUM, 396  
   float, 61, 92  
   integer, 60, 92  
   MIME danych, 242

NULL, 65, 397  
 resource, 65  
 SET, 396  
 string, 62, 93  
 TEXT, 396  
 VARBINARY, 396  
 VARCHAR, 395

typy  
 danych w kolumnach, 392  
 daty i czasu, 393  
 liczbowe, 392  
 łańcuchowe, 395  
 skalarne, 60  
 specjalne, 64  
 złożone, 64

## U

Ubuntu  
   instalacja PHP, 21  
   instalacja serwera WWW, 20  
   testowanie instalacji, 22  
   uruchamianie serwera, 22  
 uruchamianie serwera, 16, 22, 24, 27  
 XAMPP, 32  
 ustalenie haseł dostępowych, 39  
 usuwanie  
   cookie, 263  
   danych, 409  
   katalogów, 188  
   plików, 189  
   tabel, 392

## W

weryfikacja, 279  
 wiersz poleceń, 40  
 Windows  
   instalacja MySQL, 374  
   instalacja pakietu XAMPP, 35  
   instalacja PHP, 23  
   instalacja serwera WWW, 23  
   testowanie instalacji, 26  
   uruchamianie serwera, 24  
   uruchamianie serwera MySQL, 380  
   zarządzanie serwerami, 36  
 właściwości, 290  
 wprowadzanie danych, 397  
 wybór bazy danych, 386, 411  
 wyjątek PDOException, 431  
 wyjątki, 317  
   własne, 327  
 wykonywanie zapytań, 413, 419

- wylogowanie, 285
- wymuszanie konwersji, 90
- wyniki głosowania, 446, 449
- wyrażenia
  - warunkowe, 99
  - złożone, 151
- wysyłanie
  - danych, 241
  - plików, 237
  - plików do przeglądarki, 244
  - plików do serwera, 236
  - poczty, 363
  - zawartości plików, 242
- wyświetlanie
  - ciągów znaków, 152
  - daty i czasu, 134, 138
  - informacji, 51
  - list dla plików, 185
  - listy odnośników, 246
  - wartości zmiennych, 66
  - zawartości tablic, 171
  - zawartości katalogu, 183
- zmienianie wielkości liter, 158
- zmiennie, 59, 65
  - globalne, 87, 123
  - iteracyjne, 113
  - lokalne, 126
  - sesji, 271
  - statyczne, 127
  - superglobalne, 128
- znacznik
  - <form>, 236
  - czasu, 142
- znaczniki
  - kanoniczne, 48
  - PHP, 47
  - skryptów HTML, 49
  - typu ASP, 48
  - typu SGML, 48
- znak
  - apostrofu, 62
  - cudzysłowu, 63
- znaki specjalne, 69
- zwracanie wartości, 121

## X

XAMPP, 30, 380

## Z

- zagnieżdżanie
  - instrukcji warunkowych, 97
  - pętli, 113
- zamykanie plików, 199
- zapamiętywanie danych, 264
- zapis
  - danych, 207
  - plików graficznych, 340
  - w pliku tekstowym, 234
- zapytania, 397
  - aktualizujące dane, 416
  - modyfikujące dane, 422, 433, 437
  - pobierające dane, 413, 420, 428, 436
- zarządzanie
  - bazami danych, 384
  - serwerami, 36
- zasady konwersji, 91
- zasięg zmiennych, 123
- zatrzymywanie serwera, 22, 24
- zliczanie liczby odwiedzin, 267
- złożone wyrażenia warunkowe, 99
- zmiana
  - katalogu bieżącego, 188
  - nazw kolumn, 402



# PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



1. ZAREJESTRUJ SIĘ
2. PREZENTUJ KSIĄŻKI
3. ZBIERAJ PROWIZJĘ

Zmień swoją stronę WWW  
w działający bankomat!

**Dowiedz się więcej i dołącz już dzisiaj!**

<http://program-partnerski.helion.pl>



# PHP7

## Język skryptowy sam wchodzi do głowy!

Era statycznych stron internetowych minęła tak dawno, że nawet najstarsi internauci słabo ją pamiętają. Trudno byłoby dziś znaleźć w sieci stronę opartą wyłącznie na HTML-u, a gdyby nawet się to udało, można by ją chyba od razu odesłać do muzeum. Tymczasem wśród różnorodnych technik służących dynamicznemu generowaniu treści stron WWW poczesne — o ile nie najważniejsze — miejsce zajmuje PHP. Jeśli więc chcesz zacząć samodzielnie tworzyć strony, nie możesz pominąć nauki tego języka. Co więcej, musisz poznać go bardzo dobrze, ponieważ będziesz z niego korzystać na co dzień.

Ta książka pomoże Ci szybko i bezboleśnie wejść w świat języka PHP w jego najnowszej wersji. Na praktycznych przykładach nauczysz się poprawnie stosować najpierw najprostsze, a później nieco bardziej zaawansowane techniki. Zorientujesz się, co to jest i jak działa język skryptowy, jak zainstalować PHP i serwery, jak PHP współpracuje z przeglądarką internetową czy bazą danych. Nauczysz się konstruować interaktywne elementy stron WWW, budować formularze i blokować niepożądane adresy IP. Seria listingów, przejrzystych wyjaśnień i ćwiczeń do wykonania pozwoli Ci poznać PHP od podszewki i bez oporów używać go w praktyce.

## Zadbaj o profesjonalny wygląd swoich stron WWW!

**Helion** 

księgarnia internetowa



<http://helion.pl>

zamówienia telefoniczne



**0 801 339900**



**0 601 339900**

Helion SA  
ul. Kościuszki 1c, 44-100 Gliwice  
tel.: 32 230 98 63  
e-mail: [helion@helion.pl](mailto:helion@helion.pl)  
<http://helion.pl>

Sprawdź najnowsze promocje:  
● <http://helion.pl/promocje>  
Książki najchętniej czytane:  
● <http://helion.pl/bestsellery>  
Zamów informacje o nowościach:  
● <http://helion.pl/nowosci>

sięgnij po **WIĘCEJ**



KOD KORZYŚCI

ISBN 978-83-283-2478-7



9 788328 324787