

Świat zaawansowanych stron WWW  
w zasięgu Twoich rąk!

▼ Jak przygotować środowisko pracy?

▼ Jak wykorzystać technologię AJAX  
w połączeniu z PHP?

▼ Jak połączyć PHP i MySQL w jedną całość?

# PHP i MySQL

## Od podstaw

Wydanie IV

W. Jason Gilmore



Apress®

## » Idź do

- Spis treści
- Przykładowy rozdział
- Skorowidz

## » Katalog książek

- Katalog online
- Zamów drukowany katalog

## » Twój koszyk

- Dodaj do koszyka

## » Cennik i informacje

- Zamów informacje o nowościach
- Zamów cennik

## » Czytelnia

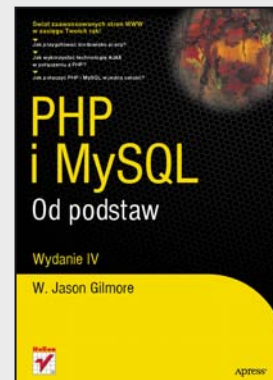
- Fragmenty książek online

## » Kontakt

Helion SA  
ul. Kościuszki 1c  
44-100 Gliwice  
tel. 32 230 98 63  
e-mail: helion@helion.pl  
© Helion 1991–2011

## PHP i MySQL. Od podstaw. Wydanie IV

Autor: W. Jason Gilmore  
Tłumaczenie: Piotr Rajca  
ISBN: 978-83-246-3056-1  
Tytuł oryginału: [Beginning PHP and MySQL:  
From Novice to Professional, Fourth Edition](#)  
Format: 172×245, stron: 668



### Świat zaawansowanych stron WWW w zasięgu Twoich rąk!

- Jak przygotować środowisko pracy?
- Jak wykorzystać technologię AJAX w połączeniu z PHP?
- Jak połączyć PHP i MySQL w jedną całość?

PHP i MySQL to duet, na którym opiera się ogromna większość ciut bardziej zaawansowanych stron w sieci. Czemu zawdzięcza on swą popularność? Niezwykłemu dopasowaniu do potrzeb, łatwej konfiguracji oraz ogromnej społeczności, zawsze chętniej do pomocy. Kiedy zaprzęgniesz ten zestaw do pracy, już po kilku godzinach zobaczysz jej pierwsze efekty!

Co zrobić, żeby osiągnąć jeszcze więcej? Nic prostszego! Sięgnij po tę książkę i zacznij swoją przygodę z PHP oraz MySQL. Na samym początku dowiesz się, jak przygotować środowisko pracy, oraz poznasz podstawy programowania w języku PHP. Potem płynnie przejdziesz do zdobywania wiedzy na temat programowania obiektowego, wyrażeń regularnych, obsługi formularzy HTML oraz integracji z usługami katalogowymi LDAP. Nauczysz się wykorzystywać mechanizm sesji, technologię AJAX oraz Zend Framework. Po przeprowadzeniu Cię przez komplet informacji na temat PHP autor przedstawi Ci bazę danych MySQL. Poznasz jej mocne elementy, sposoby podłączania klientów oraz podstawy administracji. Może w to nie wierzysz, ale po przeczytaniu tej książki nawet skomplikowane mechanizmy bazy MySQL nie będą Ci obce! Na sam koniec odkryjesz, jak wiedzę na temat PHP oraz MySQL połączyć w całość i zrealizować Twój nawet najbardziej ambitny projekt! Książka ta jest obowiązkową pozycją na półce każdego entuzjasty tworzenia dynamicznych stron WWW!

- Przygotowanie środowiska do pracy
- Podstawy PHP – typy danych, zmienne, funkcje, tablice
- Programowanie obiektowe w PHP, obsługa wyjątków i błędów
- Operacje na łańcuchach znaków, wyrażenia regularne
- Obsługa plików i korzystanie z funkcji systemu operacyjnego
- Wykorzystanie pakietów PEAR, szablony stron – pakiet Smarty
- Operacje na dacie i czasie i obsługa formularzy HTML
- Uwierzytelnianie użytkowników i zarządzanie sesją
- Integracja PHP z LDAP
- Zabezpieczanie witryn WWW
- Zastosowanie technologii AJAX z jQuery oraz wykorzystanie Zend Framework
- Możliwości bazy danych MySQL, instalacja i konfiguracja serwera MySQL
- Mechanizmy składowania i typy danych w MySQL, zabezpieczanie serwerów MySQL
- Pobieranie danych z bazy i operacje na nich w PHP
- Wykorzystanie widoków i mechanizm transakcji

**Sprawdź, jak wykorzystać możliwości PHP i MySQL  
w zakresie projektowania stron internetowych!**

# Spis treści

	<b>O autorze .....</b>	<b>19</b>
	<b>O recenzencie technicznym .....</b>	<b>21</b>
	<b>Podziękowania .....</b>	<b>23</b>
	<b>Wprowadzenie .....</b>	<b>25</b>
<b>Rozdział 1.</b>	<b>Prezentacja PHP .....</b>	<b>27</b>
	Historia .....	28
	PHP 4 .....	28
	PHP 5 .....	29
	PHP 5.3 .....	30
	PHP 6 .....	30
	Ogólne cechy języka .....	31
	Praktyczność .....	31
	Możliwości .....	32
	Potencjał .....	32
	Cena .....	33
	Podsumowanie .....	33
<b>Rozdział 2.</b>	<b>Konfiguracja środowiska .....</b>	<b>35</b>
	Wymagania wstępne instalacji .....	36
	Pobieranie serwera Apache .....	36
	Pobieranie PHP .....	36
	Pobieranie dokumentacji .....	37
	Instalacja Apache i PHP w systemie Linux .....	38
	Instalacja Apache i PHP w systemie Windows .....	39
	Instalacja serwera IIS i PHP w systemie Windows .....	41
	Testowanie instalacji .....	41
	Konfiguracja PHP .....	43
	Konfiguracja PHP w trakcie tworzenia w systemach Linux .....	43
	Modyfikowanie PHP budowanego w systemie Windows .....	44
	Konfiguracja PHP w trakcie jego działania .....	44
	Korzystanie z dyrektyw konfiguracyjnych PHP .....	44
	Dyrektywy konfiguracyjne PHP .....	46

Wybór edytora .....	59
Adobe Dreamweaver CS5 .....	59
Notepad++ .....	60
PDT (PHP Development Tools) .....	60
Zend Studio .....	60
Wybór firmy udostępniającej serwery WWW .....	60
Siedem pytań do firm udostępniających serwery WWW .....	61
Podsumowanie .....	62
<b>Rozdział 3. Podstawy PHP .....</b>	<b>63</b>
Umieszczanie kodu PHP na stronach WWW .....	63
Domyślna składnia .....	64
Krótkie znaczniki .....	64
Skrypt .....	65
Składnia ASP .....	65
Osadzanie wielu fragmentów kodu .....	65
Komentowanie kodu .....	66
Jednowierszowe komentarze w stylu C++ .....	66
Składnia skryptów powłoki .....	66
Komentarze wielowierszowe, składnia C .....	67
Generowanie danych wyjściowych .....	67
Instrukcja print() .....	67
Instrukcja echo() .....	68
Instrukcja printf() .....	69
Instrukcja sprintf() .....	70
Typy danych dostępne w PHP .....	70
Skalarne typy danych .....	70
Złożone typy danych .....	72
Konwersja typów danych przy użyciu rzutowania .....	73
Adaptacja typów danych poprzez ich zmienianie .....	74
Funkcje związane z typami danych .....	75
Funkcje identyfikujące typ .....	75
Identyfikatory .....	76
Zmienne .....	76
Deklaracje zmiennych .....	76
Zasięg zmiennych .....	78
Zmienne superglobalne .....	80
Stałe .....	84
Definiowanie stałych .....	85
Wyrażenia .....	85
Operandy .....	85
Operatory .....	85
Umieszczanie wartości w łańcuchach znaków .....	91
Cudzysłowy .....	91
Sekwencje sterujące .....	91
Apostrofy .....	92
Nawiasy klamrowe .....	92

Składnia heredoc .....	93
Składnia nowdoc .....	93
Struktury sterujące .....	93
Instrukcje warunkowe .....	94
Pętle .....	96
Instrukcje dołączania plików .....	101
Podsumowanie .....	103
<b>Rozdział 4. Funkcje .....</b>	<b>105</b>
Wywoływanie funkcji .....	105
Tworzenie funkcji .....	106
Przekazywanie argumentów przez wartość .....	107
Przekazywanie przez referencję .....	107
Domyślne wartości argumentów .....	108
Stosowanie informacji o typie .....	109
Zwracanie wartości z funkcji .....	109
Funkcje rekurencyjne .....	110
Biblioteki funkcji .....	113
Podsumowanie .....	113
<b>Rozdział 5. Tablice .....</b>	<b>115</b>
Czym jest tablica? .....	115
Tworzenie tablic .....	116
Tworzenie tablic przy użyciu array() .....	117
Odczytywanie elementów tablicy przy użyciu list() .....	118
Zapisywanie w tablicy predefiniowanego zakresu wartości .....	118
Sprawdzanie, czy zmienna jest tablicą .....	119
Wyświetlanie zawartości tablic .....	119
Wyświetlanie tablic w ramach testowania skryptu .....	120
Dodawanie i usuwanie elementów tablic .....	121
Dodawanie wartości na początku tablicy .....	121
Dodawanie elementów na końcu tablicy .....	121
Usuwanie elementu z początku tablicy .....	121
Odnajdywanie elementów w tablicach .....	122
Przeszukiwanie tablic .....	122
Przeszukiwanie tablic z kluczami asocjacyjnymi .....	122
Poszukiwanie wartości w tablicach asocjacyjnych .....	123
Pobieranie kluczy tablicy .....	123
Pobieranie wartości z tablicy .....	123
Przeglądanie tablic .....	124
Pobieranie klucza aktualnego elementu tablicy .....	124
Pobieranie wartości aktualnego elementu tablicy .....	124
Pobieranie aktualnego klucza i wartości .....	125
Przesuwanie wskaźnika tablicy .....	125
Przekazywanie wartości z tablicy do funkcji .....	126
Określanie wielkości oraz unikalności tablicy .....	127
Określanie wielkości tablicy .....	127
Określanie częstotliwości występowania wartości .....	127
Określanie unikalnych elementów tablicy .....	128

Sortowanie tablic .....	128
Zmiana kolejności elementów tablicy .....	129
Zamiana kluczy i wartości .....	129
Sortowanie tablic .....	129
Łączenie i dzielenie tablic na różne sposoby .....	134
Łączenie tablic .....	134
Rekurencyjne łączenie tablic .....	134
Łączenie dwóch tablic .....	135
Pobieranie fragmentu tablicy .....	135
Wycinanie elementów z tablicy .....	136
Wyznaczanie części wspólnej tablic .....	136
Określanie części wspólnej tablic asocjacyjnych .....	137
Określanie różnicy tablic .....	137
Określanie różnicy tablic asocjacyjnych .....	138
Inne przydatne funkcje operujące na tablicach .....	139
Zwracanie losowego zbioru kluczy .....	139
Losowa zmiana kolejności elementów tablicy .....	139
Dodawanie wartości tablicy .....	139
Dzielenie tablicy .....	140
Podsumowanie .....	140
<b>Rozdział 6. Programowanie obiektowe .....</b>	<b>141</b>
Zalety programowania obiektowego .....	141
Hermetyzacja .....	141
Dziedziczenie .....	142
Polimorfizm .....	142
Kluczowe pojęcia programowania obiektowego .....	143
Klasy .....	143
Obiekty .....	144
Właściwości .....	144
Stałe .....	149
Metody .....	150
Konstruktory i destruktory .....	153
Konstruktory .....	153
Destruktory .....	156
Składowe statyczne .....	156
Słowo kluczowe instanceof .....	157
Funkcje pomocnicze .....	158
Sprawdzanie, czy istnieje metoda .....	159
Automatyczne wczytywanie klas .....	159
Podsumowanie .....	160
<b>Rozdział 7. Zaawansowane zagadnienia programowania obiektowego .....</b>	<b>161</b>
Zaawansowane mechanizmy obiektowe niedostępne w PHP .....	162
Klonowanie obiektów .....	162
Przykład klonowania obiektu .....	162
Metoda __clone() .....	163

Dziedziczenie .....	164
Dziedziczenie klas .....	165
Dziedziczenie i konstruktory .....	166
Dziedziczenie i późne wiązanie statyczne .....	168
Interfejsy .....	168
Implementacja pojedynczego interfejsu .....	170
Implementacja kilku interfejsów .....	170
Klasy abstrakcyjne .....	171
Prezentacja przestrzeni nazw .....	172
Podsumowanie .....	174
<b>Rozdział 8. Obsługa błędów i wyjątków .....</b>	<b>175</b>
Dyrektywy konfiguracyjne .....	176
Rejestracja błędów .....	178
Obsługa wyjątków .....	181
Dlaczego obsługa wyjątków jest wygodna? .....	181
Implementacja obsługi wyjątków w języku PHP .....	182
Wyjątki SPL .....	186
Podsumowanie .....	187
<b>Rozdział 9. Łańcuchy znaków i wyrażenia regularne .....</b>	<b>189</b>
Wyrażenia regularne .....	190
Składnia wyrażeń regularnych (w stylu POSIX) .....	190
Funkcje obsługi wyrażeń regularnych (rozszerzony standard POSIX) .....	192
Składnia wyrażeń regularnych (zgodnych z Perl) .....	195
Inne funkcje operujące na łańcuchach znaków .....	201
Określanie długości łańcucha znaków .....	202
Porównywanie dwóch łańcuchów znaków .....	202
Zmiana wielkości liter .....	204
Konwersja łańcucha znaków na kod HTML i na odwrót .....	206
Alternatywy dla funkcji używających wyrażeń regularnych .....	210
Dopełnianie oraz przycinanie łańcuchów znaków .....	216
Zliczanie znaków i słów .....	217
Stosowanie PEAR — pakiet Validate_US .....	219
Instalowanie pakietu Validate_US .....	220
Stosowanie pakietu Validate_US .....	220
Podsumowanie .....	221
<b>Rozdział 10. Obsługa plików i korzystanie z systemu operacyjnego .....</b>	<b>223</b>
Pobieranie informacji o plikach i katalogach .....	224
Przetwarzanie ścieżki .....	224
Określanie wielkości pliku, katalogu i dysku .....	225
Określanie czasu dostępu i modyfikacji .....	228
Operacje na plikach .....	229
Pojęcie zasobu .....	229
Rozpoznawanie znaków nowego wiersza .....	229
Rozpoznawanie znaku końca pliku .....	229
Otwieranie i zamykanie pliku .....	230
Odczytywanie zawartości plików .....	231

Przesuwanie wskaźnika pliku .....	237
Odczytywanie zawartości katalogu .....	238
Wykonywanie poleceń systemowych .....	239
Usuwanie katalogu .....	239
Wykonywanie programów z poziomu systemu operacyjnego .....	240
Zabezpieczanie danych wejściowych .....	241
Funkcje do wykonywania programów .....	242
Podsumowanie .....	244
<b>Rozdział 11. PEAR .....</b>	<b>245</b>
Potęga PEAR — konwersja formatów liczbowych .....	246
Instalacja i aktualizacja PEAR .....	246
Instalacja PEAR .....	247
PEAR i firmy udostępniające serwery .....	248
Aktualizacja PEAR .....	248
Korzystanie z menedżera pakietów PEAR .....	248
Przeglądanie zainstalowanych pakietów PEAR .....	249
Pobieranie informacji o zainstalowanych pakietach .....	249
Instalacja wybranego pakietu PEAR .....	250
Dołączanie pakietów do skryptów .....	251
Aktualizacja pakietów .....	252
Odinstalowanie pakietu .....	253
Użycie wcześniejszej wersji pakietu .....	253
Menedżer pakietów Pyrus .....	253
Instalacja menedżera Pyrus .....	253
Podsumowanie .....	254
<b>Rozdział 12. Data i czas .....</b>	<b>255</b>
Uniksowy znacznik czasu .....	255
Biblioteka funkcji do obsługi dat i czasu .....	256
Weryfikacja dat .....	256
Formatowanie dat i czasu .....	256
Konwersja znacznika czasu na zrozumiałą postać .....	259
Posługiwanie się znacznikami czasu .....	260
Sztuka czasu .....	261
Wyświetlanie zlokalizowanych dat i czasu .....	261
Wyświetlanie daty ostatniej modyfikacji witryny .....	265
Określanie liczby dni w bieżącym miesiącu .....	265
Określanie liczby dni w podanym miesiącu .....	265
Wyliczanie daty na podstawie różnicy w dniach .....	266
Usprawnienia obsługi dat i czasu w PHP 5.1+ .....	266
Konstruktor klasy DateTime .....	267
Formatowanie dat .....	267
Określanie daty po utworzeniu obiektu .....	267
Określanie czasu po utworzeniu obiektu .....	268
Modyfikowanie dat i czasu .....	268
Obliczanie różnicy pomiędzy dwiema datami .....	268
Podsumowanie .....	269



<b>Rozdział 13. Obsługa formularzy HTML .....</b>	<b>271</b>
PHP i formularze HTML .....	271
Prosty przykład .....	272
Weryfikacja danych z formularzy .....	273
Usuwanie plików .....	273
Cross-site scripting .....	274
Zabezpieczanie danych wprowadzanych przez użytkowników .....	275
Weryfikacja i zabezpieczanie danych przy użyciu rozszerzenia Filter .....	277
Korzystanie ze złożonych komponentów formularzy .....	278
Wykorzystanie PEAR — HTML_QuickForm2 .....	280
Instalacja pakietu HTML_QuickForm2 .....	281
Tworzenie i weryfikacja danych prostego formularza .....	281
Podsumowanie .....	283
<b>Rozdział 14. Uwierzytelnianie użytkowników .....</b>	<b>285</b>
Uwierzytelnianie w oparciu o protokół HTTP .....	285
Korzystanie z plików .htaccess serwera Apache .....	286
Uwierzytelnianie przy użyciu możliwości PHP .....	287
Zmienne PHP związane z uwierzytelnianiem .....	287
Użyteczne funkcje .....	287
Uwierzytelnianie w oparciu o stałe dane .....	289
Uwierzytelnianie w oparciu o pliki .....	289
Uwierzytelnianie z wykorzystaniem bazy danych .....	291
Wykorzystanie możliwości PEAR — pakiet Auth_HTTP .....	292
Zarządzanie danymi uwierzytelniającymi .....	294
Testowanie siły hasła przy użyciu biblioteki CrackLib .....	294
Odzyskiwanie haseł przy użyciu jednorazowych adresów URL .....	296
Podsumowanie .....	298
<b>Rozdział 15. Obsługa przesyłania plików na serwer .....</b>	<b>299</b>
Przesyłanie plików przy użyciu protokołu HTTP .....	299
Obsługa przesyłanych plików przy użyciu PHP .....	300
Dyrektywy konfiguracyjne związane z przesyłaniem plików .....	300
Tablica \$_FILES .....	302
Funkcje PHP do obsługi przesyłanych plików .....	302
Komunikaty błędów .....	304
Prosty przykład .....	304
Wykorzystanie PEAR — HTTP_Upload .....	305
Instalacja pakietu HTTP_Upload .....	305
Przesyłanie pliku na serwer .....	306
Uzyskiwanie informacji o przesłanym pliku .....	306
Obsługa przesyłania większej liczby plików .....	307
Podsumowanie .....	308
<b>Rozdział 16. Zagadnienia sieciowe .....</b>	<b>309</b>
DNS, usługi i serwery .....	310
DNS .....	310
Usługi .....	313
Tworzenie połączeń używających gniazd .....	314

Poczta elektroniczna .....	316
Dyrektywy konfiguracyjne .....	316
Wysyłanie wiadomości ze skryptów PHP .....	317
Popularne operacje sieciowe .....	321
Testowanie połączenia z serwerem .....	321
Tworzenie skanera portów .....	322
Tworzenie konwertera podsieci .....	322
Testowanie przepustowości łącza użytkownika .....	324
Podsumowanie .....	324
<b>Rozdział 17. PHP i LDAP .....</b>	<b>325</b>
Stosowanie LDAP w języku PHP .....	326
Konfiguracja LDAP w PHP .....	326
Nawiązywanie połączenia z serwerem LDAP .....	326
Pobieranie danych z serwera LDAP .....	329
Określanie liczby zwróconych rekordów .....	331
Sortowanie rekordów LDAP .....	332
Wstawianie danych LDAP .....	332
Aktualizacja danych LDAP .....	334
Usuwanie danych z serwera LDAP .....	334
Operacje na rozpoznawalnych nazwach .....	335
Obsługa błędów .....	336
Podsumowanie .....	337
<b>Rozdział 18. Obsługa sesji .....</b>	<b>339</b>
Czym jest obsługa sesji? .....	339
Proces obsługi sesji .....	340
Dyrektywy konfiguracyjne .....	340
Zarządzanie sposobem przechowywania danych sesyjnych .....	340
Określanie ścieżki do plików sesji .....	341
Automatyczne włączanie sesji .....	341
Określanie nazwy sesji .....	342
Wybór ciasteczek lub przepisywania adresów URL .....	342
Automatyczne przepisywanie adresów URL .....	342
Określanie czasu istnienia ciasteczka .....	342
Określanie ścieżki ciasteczka .....	342
Określanie katalogów do przechowywania stron wykorzystujących sesje .....	343
Korzystanie z sesji .....	344
Rozpoczynanie sesji .....	344
Usuwanie sesji .....	345
Ustawianie i pobieranie identyfikatora sesji .....	345
Tworzenie i usuwanie zmiennych sesyjnych .....	346
Kodowanie i dekodowanie danych sesyjnych .....	346
Praktyczne przykłady stosowania sesji .....	348
Automatyczne logowanie powracających użytkowników .....	348
Generowanie listy ostatnio odwiedzonych stron .....	350
Tworzenie własnych procedur obsługi sesji .....	351
Określanie funkcji stosowanych w obsłudze sesji .....	352
Obsługa sesji z użyciem bazy danych MySQL .....	352
Podsumowanie .....	355

<b>Rozdział 19. Stosowanie szablonów z pakietem Smarty</b> .....	<b>357</b>
Czym jest mechanizm obsługi szablonów? .....	357
Przedstawienie mechanizmu Smarty .....	359
Instalacja Smarty .....	360
Korzystanie ze Smarty .....	361
Logika prezentacji w szablonach Smarty .....	363
Komentarze .....	363
Modyfikatory zmiennych .....	363
Struktury sterujące .....	366
Instrukcje .....	369
Tworzenie plików konfiguracyjnych .....	371
Korzystanie ze zmiennych konfiguracyjnych .....	372
Stosowanie arkuszy stylów w szablonach Smarty .....	373
Wykorzystanie pamięci podręcznej .....	374
Określanie czasu przechowywania stron .....	374
Eliminowanie narzutów dzięki użyciu metody isCached() .....	375
Przechowywanie w pamięci podręcznej wielu wersji tego samego szablonu .....	375
Kilka ostatnich uwag o stosowaniu pamięci podręcznej .....	376
Podsumowanie .....	377
<b>Rozdział 20. Usługi sieciowe</b> .....	<b>379</b>
Dlaczego powstały usługi sieciowe? .....	380
RSS .....	381
Format RSS .....	382
Prezentacja SimplePie .....	383
Instalacja SimplePie .....	384
Przetwarzanie kanału RSS przy użyciu SimplePie .....	384
Przetwarzanie większej liczby kanałów .....	386
SimpleXML .....	387
Wczytywanie dokumentów XML .....	387
Analiza XML .....	389
Podsumowanie .....	391
<b>Rozdział 21. Zabezpieczanie witryn WWW</b> .....	<b>393</b>
Bezpieczna konfiguracja PHP .....	394
Dyrektywy konfiguracyjne związane z bezpieczeństwem .....	394
Ukrywanie szczegółów konfiguracji .....	395
Ukrywanie serwera Apache .....	396
Ukrywanie PHP .....	396
Ukrywanie wrażliwych danych .....	398
Ukrywanie głównego katalogu dokumentów .....	398
Zabronienie dostępu do plików z określonymi rozszerzeniami .....	398
Szyfrowanie danych .....	399
Funkcje szyfrujące PHP .....	399
Pakiet MCRYPT .....	400
Podsumowanie .....	401

<b>Rozdział 22. Korzystanie z technologii Ajax przy użyciu jQuery i PHP .....</b>	<b>403</b>
Przedstawienie Ajaksa .....	403
Prezentacja jQuery .....	405
Instalacja jQuery .....	405
Prosty przykład .....	405
Odpowiadanie na zdarzenia .....	406
Biblioteka jQuery i DOM .....	407
Mechanizm weryfikacji dostępności nazwy użytkownika .....	409
Określanie, czy nazwa użytkownika jest dostępna .....	410
Podsumowanie .....	412
<b>Rozdział 23. Tworzenie witryn dla odbiorców z całego świata .....</b>	<b>415</b>
Tłumaczenie witryn przy użyciu pakietu gettext .....	416
Etap 1. Aktualizacja skryptów .....	416
Etap 2. Tworzenie repozytorium lokalizacji .....	417
Etap 3. Tworzenie plików tłumaczeń .....	418
Etap 4. Przetłumaczenie tekstów .....	418
Etap 5. Generowanie plików binarnych .....	419
Etap 6. Określanie odpowiedniego języka w skryptach .....	419
Lokalizacja dat, liczb i godzin .....	420
Podsumowanie .....	421
<b>Rozdział 24. Wprowadzenie do stosowania Zend Framework .....</b>	<b>423</b>
Przedstawienie wzorca MVC .....	423
Szkielety aplikacji PHP .....	425
CakePHP .....	426
Solar .....	426
Symfony .....	426
Zend Framework .....	426
Przedstawienie Zend Framework .....	427
Instalacja Zend Framework .....	428
Tworzenie pierwszej witryny używającej Zend Framework .....	428
Podsumowanie .....	439
<b>Rozdział 25. Prezentacja bazy danych MySQL .....</b>	<b>441</b>
Dlaczego MySQL jest tak popularny? .....	441
Elastyczność .....	442
Moc .....	442
Elastyczne opcje licencyjne .....	444
(Hiper)aktywna społeczność użytkowników .....	445
Ewolucja serwera MySQL .....	445
MySQL 4 .....	445
MySQL 5 .....	446
MySQL 5.1 .....	447
MySQL 5.4 oraz 5.5 .....	447
Najważniejsi użytkownicy serwerów MySQL .....	448
craigslist .....	448
Wikipedia .....	448
Inni znaczący użytkownicy .....	448
Podsumowanie .....	448

<b>Rozdział 26. Instalacja i konfiguracja serwera MySQL .....</b>	<b>451</b>
Pobieranie serwera MySQL .....	451
Instalacja MySQL .....	452
Instalacja serwera MySQL w systemie Linux .....	452
Instalacja i konfiguracja MySQL w systemie Windows .....	455
Określanie hasła administratora .....	457
Uruchamianie i zatrzymywanie serwera MySQL .....	458
Ręczna kontrola pracy procesu .....	458
Konfiguracja i optymalizacja serwera MySQL .....	459
Skrypt mysqld_safe .....	459
Parametry konfiguracyjne i optymalizacyjne .....	460
Plik my.cnf .....	463
Konfiguracja PHP pod kątem współpracy z MySQL .....	465
Rekonfiguracja PHP w systemie Linux .....	465
Rekonfiguracja PHP w systemie Windows .....	465
Podsumowanie .....	466
<b>Rozdział 27. Wiele klientów MySQL .....</b>	<b>467</b>
Prezentacja klientów obsługiwanych z wiersza poleceń .....	467
Klient mysql .....	467
Klient mysqladmin .....	475
Inne przydatne klienty .....	477
Opcje klientów .....	480
Klienty MySQL z graficznym interfejsem użytkownika .....	482
Aplikacja phpMyAdmin .....	483
Podsumowanie .....	484
<b>Rozdział 28. Mechanizmy składowania i typy danych MySQL .....</b>	<b>485</b>
Mechanizmy składowania .....	485
MyISAM .....	486
IBMDB2I .....	488
InnoDB .....	488
MEMORY .....	489
MERGE .....	490
FEDERATED .....	490
ARCHIVE .....	492
CSV .....	492
EXAMPLE .....	492
BLACKHOLE .....	492
Pytania i odpowiedzi dotyczące mechanizmów składowania .....	493
Typy danych i atrybuty .....	494
Typy danych .....	494
Operacje na bazach danych i tabelach .....	502
Operacje na bazach danych .....	502
Operacje na tabelach .....	504
Modyfikowanie struktury tabel .....	506
Baza danych INFORMATION_SCHEMA .....	507
Podsumowanie .....	509

<b>Rozdział 29. Zabezpieczanie serwerów MySQL .....</b>	<b>511</b>
Co należy zrobić na samym początku? .....	512
Zabezpieczanie procesu serwera MySQL .....	513
System uprawnień serwera MySQL .....	513
Sposób działania systemu uprawnień .....	514
Gdzie są przechowywane informacje o uprawnieniach? .....	516
Zarządzanie użytkownikami i uprawnieniami .....	523
Tworzenie użytkowników .....	524
Usuwanie użytkowników .....	524
Zmiana nazwy istniejących użytkowników .....	524
Polecenia GRANT i REVOKE .....	525
Przeglądanie posiadanych uprawnień .....	530
Ograniczanie dostępu do zasobów .....	530
Bezpieczne połączenia z serwerem MySQL .....	531
Opcje polecenia GRANT .....	531
Opcje SSL .....	533
Uruchamianie serwera MySQL obsługującego bezpieczne połączenia .....	534
Nawiązywanie połączenia przy użyciu klienta obsługującego SSL .....	534
Określanie opcji SSL w pliku konfiguracyjnym my.cnf .....	534
Podsumowanie .....	535
<b>Rozdział 30. Współpraca PHP i MySQL .....</b>	<b>537</b>
Wymagania instalacji .....	538
Włączanie rozszerzenia mysqli w systemach Linux oraz Unix .....	538
Włączanie rozszerzenia mysqli w systemie Windows .....	538
Stosowanie sterownika MySQL .....	538
Zarządzanie uprawnieniami użytkowników .....	539
Korzystanie z przykładowych danych .....	539
Praktyczne wykorzystanie rozszerzenia mysqli .....	540
Konfiguracja i nawiązywanie połączenia .....	540
Obsługa błędów połączenia .....	541
Pobieranie informacji o błędach .....	541
Przechowywanie informacji o połączeniu w osobnym pliku .....	542
Operacje na bazie danych .....	543
Przesyłanie zapytań do bazy danych .....	543
Przetwarzanie wyników zapytania .....	546
Określanie liczby zwróconych oraz zmodyfikowanych wierszy .....	547
Korzystanie z poleceń przygotowanych .....	548
Stosowanie transakcji .....	552
Włączanie trybu automatycznego zatwierdzania .....	553
Zatwierdzanie transakcji .....	553
Wycofywanie transakcji .....	553
Podsumowanie .....	553
<b>Rozdział 31. Prezentacja PDO .....</b>	<b>555</b>
Kolejna warstwa abstrakcji bazy danych? .....	556
Stosowanie PDO .....	557
Instalacja PDO .....	557
Obsługiwane bazy danych .....	558

Nawiązywanie połączenia z serwerem i wybór bazy danych .....	558
Obsługa błędów .....	560
Pobieranie i ustawianie atrybutów .....	562
Wykonywanie zapytań .....	563
Prezentacja poleceń przygotowanych .....	564
Pobieranie danych .....	567
Tworzenie powiązanych kolumn .....	569
Stosowanie transakcji .....	570
Podsumowanie .....	571
<b>Rozdział 32. Podprogramy składowane .....</b>	<b>573</b>
Czy warto używać podprogramów składowanych? .....	574
Zalety podprogramów składowanych .....	574
Wady podprogramów składowanych .....	574
Implementacja podprogramów składowanych w serwerze MySQL .....	575
Tworzenie podprogramów składowanych .....	575
Deklarowanie i ustawianie zmiennych .....	578
Wykonywanie podprogramu składowanego .....	579
Tworzenie i stosowanie złożonych podprogramów składowanych .....	579
Wywoływanie podprogramu z innego podprogramu .....	586
Modyfikowanie podprogramów składowanych .....	586
Usuwanie podprogramów składowanych .....	586
Wyświetlanie informacji o statusie podprogramu .....	587
Wyświetlanie polecenia użytego do utworzenia podprogramu .....	588
Obsługa warunków .....	588
Integracja podprogramów składowanych w aplikacjach internetowych .....	589
Strona do wyliczania i prezentacji premii .....	589
Pobieranie wielu wierszy wyników .....	590
Podsumowanie .....	590
<b>Rozdział 33. Wyzwalacze .....</b>	<b>591</b>
Prezentacja wyzwalaczy .....	591
Jakie są zastosowania wyzwalaczy? .....	592
Wykonywanie akcji przed zdarzeniem .....	592
Wykonywanie akcji po zdarzeniu .....	592
Wyzwalacze wykonywane przed zdarzeniem (BEFORE) i po nim (AFTER) .....	593
Obsługa wyzwalaczy w serwerze MySQL .....	594
Tworzenie wyzwalaczy .....	594
Wyświetlanie istniejących wyzwalaczy .....	596
Modyfikacja wyzwalaczy .....	598
Usuwanie wyzwalaczy .....	598
Wykorzystanie wyzwalaczy w aplikacjach internetowych .....	598
Podsumowanie .....	599
<b>Rozdział 34. Widoki .....</b>	<b>601</b>
Przedstawienie widoków .....	602
Korzystanie z widoków w serwerze MySQL .....	602
Tworzenie i wykonywanie widoków .....	602
Wyświetlanie informacji o widokach .....	607

Modyfikowanie widoków .....	609
Usuwanie widoków .....	609
Widoki aktualizujące.....	609
Stosowanie widoków w aplikacjach internetowych.....	610
Podsumowanie .....	611
<b>Rozdział 35. Praktyczne przykłady stosowania zapytań .....</b>	<b>613</b>
Przykładowe dane .....	614
Tabelaryczna prezentacja danych przy wykorzystaniu PEAR .....	614
Instalacja pakietu HTML_Table .....	615
Tworzenie prostej tabeli .....	615
Tworzenie czytelniejszych wierszy .....	616
Generowanie tabel na podstawie informacji z bazy danych .....	617
Sortowanie wyników .....	618
Podział wyników na strony .....	619
Wyświetlanie numerów stron .....	621
Pobieranie danych z wielu tabel przy użyciu podzapytań .....	623
Porównywanie przy użyciu podzapytań .....	624
Stosowanie podzapytań do sprawdzania istnienia danych .....	624
Pielęgnacja baz danych przy użyciu podzapytań .....	625
Stosowanie podzapytań w skryptach PHP .....	625
Przeglądanie wyników przy wykorzystaniu kursora .....	626
Podstawowe informacje o kursorach .....	626
Tworzenie kursora .....	627
Otwieranie kursora .....	627
Korzystanie z kursora .....	627
Zamykanie kursora .....	628
Stosowanie kursorów w skryptach PHP .....	628
Podsumowanie .....	629
<b>Rozdział 36. Indeksacja i wyszukiwanie .....</b>	<b>631</b>
Indeksacja baz danych .....	631
Indeks główny .....	632
Indeksy unikalne .....	633
Indeksy normalne .....	634
Indeksy pełnotekstowe .....	635
Indeksacja — najlepsze praktyki .....	639
Wyszukiwanie z użyciem formularzy HTML .....	639
Realizacja prostego wyszukiwania .....	639
Rozszerzanie możliwości wyszukiwania .....	640
Stosowanie wyszukiwania pełnotekstowego .....	642
Podsumowanie .....	643
<b>Rozdział 37. Transakcje .....</b>	<b>645</b>
Co to jest transakcja? .....	645
Możliwości stosowania transakcji na serwerze MySQL .....	646
Wymagania systemowe .....	646
Tworzenie tabeli .....	646



Przykładowy projekt .....	647
Tworzenie tabel i przykładowych danych .....	647
Realizacja przykładowej transakcji .....	648
Rady dotyczące korzystania z transakcji .....	650
Tworzenie aplikacji transakcyjnych w PHP .....	650
Modyfikacja wyprzedaży .....	650
Podsumowanie .....	652
<b>Rozdział 38. Importowanie i eksportowanie danych .....</b>	<b>653</b>
Przykładowa tabela .....	653
Separacja poszczególnych elementów danych .....	654
Importowanie danych .....	654
Importowanie danych przy użyciu polecenia LOAD DATA INFILE .....	655
Importowanie danych przy użyciu klienta mysqlimport .....	658
Wczytywanie danych przy użyciu skryptu PHP .....	660
Eksportowanie danych .....	661
Polecenie SELECT INTO OUTFILE .....	661
Podsumowanie .....	664
<b>Skorowidz .....</b>	<b>665</b>

## ROZDZIAŁ 2



# Konfiguracja środowiska

Czytelnik ma zapewne zamiar uruchamiać swoje aplikacje, używając istniejącej infrastruktury lub korzystając z usług firmy zajmującej się prowadzeniem i udostępnianiem serwerów WWW. W ten sposób można uniknąć konieczności dogłębnego poznawania tajników konfiguracji i administracji serwerów WWW. Niemniej większość programistów preferuje tworzenie aplikacji na lokalnym komputerze lub laptopie bądź na dedykowanym serwerze. Można zatem przypuszczać, że znajomość instalacji PHP i serwera WWW (w naszym przypadku będą to serwery Apache i IIS) i tak będzie konieczna.

Podstawowa znajomość tego procesu ma także jeszcze jedną zaletę: zapewnia możliwość dokładniejszego poznania wielu możliwości i cech PHP oraz serwera WWW, o których zapewne nie wspomniano by w większości wstępnych prezentacji języka. Wiedza ta może się okazać bardzo przydatna podczas określania, czy środowisko serwera spełnia wymagania konkretnego projektu, oraz podczas rozwiązywania problemów, jakie mogą się pojawić podczas instalowania dodatkowego oprogramowania, a mogą wynikać z niewłaściwej lub błędnej instalacji PHP.

W tym rozdziale Czytelnik przejdzie proces instalacji PHP na serwerach działających zarówno w systemach Windows, jak i Linux. Ponieważ w przypadku braku serwera WWW PHP jest raczej bezużyteczne, zostaną także przedstawione sposoby instalacji serwera Apache w systemach Windows i Linux oraz serwera IIS w systemie Windows.

Pod koniec rozdziału zamieszczono informacje dotyczące edytorów oraz zintegrowanych środowisk programistycznych (IDE), umożliwiających pisanie programów w języku PHP. Podałem także listę kluczowych pytań, jakie należy zadać każdej firmie udostępniającej serwery WWW, z której usług potencjalnie chcemy skorzystać.

Konkretnie rzecz ujmując, w tym rozdziale zostały opisane następujące zagadnienia:

- Instalacja serwera Apache i języka PHP w systemie Linux.
- Instalacja serwerów Apache i IIS oraz języka PHP w systemie Windows.
- Testowanie instalacji w celu sprawdzenia, czy wszystkie komponenty środowiska dobrze działają, oraz sposoby rozwiązywania najczęściej występujących problemów.
- Konfiguracja PHP w celu zaspokojenia praktycznie wszystkich wymagań, jakie można by sobie wyobrazić.
- Wybór odpowiedniego środowiska programistycznego, które sprawi, że pisanie kodu PHP będzie szybsze i bardziej wydajne.
- Wybór firmy udostępniającej serwery WWW, której oferta spełnia konkretne wymagania.

## Wymagania wstępne instalacji

Zacznijmy proces instalacji od pobrania niezbędnego oprogramowania. W najprostszym przypadku konieczne będzie pobranie języka PHP oraz odpowiedniego serwera WWW (w zależności od używanej platformy systemowej i preferencji będzie to serwer Apache lub IIS 7). Jeśli używana platforma systemowa zmusza do pobrania dodatkowego oprogramowania, to informacje na ten temat zostaną podane w odpowiedniej części rozdziału.

- 
- **Wskazówka** W tym rozdziale Czytelnik zostanie przeprowadzony przez proces ręcznej instalacji i konfiguracji niezbędnego środowiska. Samodzielna instalacja PHP i serwera Apache jest dobrym pomysłem, gdyż umożliwiła poznanie wielu dostępnych opcji konfiguracyjnych, co z kolei pozwala uzyskać większą kontrolę nad sposobem działania witryny. Jeśli jednak Czytelnik i tak ma zamiar skorzystać z usług firmy prowadzącej serwery WWW i zależy mu na szybkim uruchomieniu środowiska oraz zabranii się do pracy, to warto rozważyć zastosowanie pakietu XAMPP ([www.apachefriends.org/en/xampp.html](http://www.apachefriends.org/en/xampp.html)) — bezpłatnej, zautomatyzowanej instalacji zawierającej serwer Apache oraz kilka innych komponentów, takich jak PHP, Perl oraz MySQL. XAMPP jest dostępny w wersjach przeznaczonych dla systemów Windows oraz Linux, a aktualnie trwają prace nad wersjami dla systemów Mac OS X oraz Solaris. Jeśli Czytelnik planuje skorzystać z PHP wraz z serwerem IIS, sugeruję postępować zgodnie z informacjami zamieszczonymi w podrozdziale „Instalacja serwera IIS i PHP w systemie Windows”.
- 

## Pobieranie serwera Apache

Obecnie pakiety z serwerem Apache są dostępne we wszystkich najpopularniejszych dystrybucjach systemu Linux. Jeśli zatem Czytelnik używa jednej z tych platform systemowych, to istnieje całkiem duże prawdopodobieństwo, że serwer ten już jest zainstalowany na jego komputerze, a jeśli nie, to można to łatwo zrobić, korzystając z systemu zarządzania pakietami używanego w danej dystrybucji Linuksa (na przykład w systemie Ubuntu będzie to polecenie `apt-get`). Jeśli Czytelnik używa systemu Mac OS X, to serwer Apache jest w nim instalowany domyślnie. W każdym z tych przypadków można przejść bezpośrednio do kolejnego podrozdziału, „Pobieranie PHP”. Jeśli jednak Czytelnik zdecyduje się samodzielnie zainstalować serwer Apache, to proszę czytać dalej.

Ze względu na ogromne obciążenie zaleca się, by podczas pobierania serwera Apache skorzystać z jego kopii (ang. *mirror*) zlokalizowanej możliwie jak najbliżej naszego miejsca pobytu. Samo wyświetlenie strony o adresie <http://httpd.apache.org/download.cgi> spowoduje, że witryna Apache spróbuje samodzielnie określić najbliższy serwer. Sugeruję wybór najnowszej stabilnej wersji serwera Apache, co umożliwi pobranie jego kodów źródłowych w formie plików `tar.gz` lub `bz2` bądź pliku binarnego przygotowanego w kilku wersjach przeznaczonych dla różnych systemów operacyjnych. Jeśli Czytelnik używa systemu Linux i planuje przygotowanie serwera ze źródeł, to powinien pobrać archiwum z jego kodami.

Jeśli Czytelnik używa systemu Windows i chciałby zainstalować na nim serwer Apache, powinien pobrać jego najnowszą stabilną, binarną wersję z katalogu `binaries/win32`. Dostępne są dwie binarne wersje serwera — wyposażona w mechanizmy obsługi SSL oraz pozbawiona tych możliwości. Obie wersje posiadają adekwatne nazwy, które ułatwią ich rozróżnienie. Na potrzeby środowiska używanego do tworzenia aplikacji PHP sugeruję pobranie wersji bez obsługi SSL.

## Pobieranie PHP

Podobnie jak Apache, także język PHP jest obecnie dostępny we wszystkich dystrybucjach systemu Linux, a w systemie Mac OS X jest instalowany domyślnie. W tych przypadkach sugeruję, by Czytelnik postępował zgodnie z wytycznymi dotyczącymi instalacji i konfiguracji środowiska, dostosowanymi do używanego systemu operacyjnego. W pozostałych przypadkach należy pobrać najnowszą stabilną wersję PHP, klikając łącze `Downloads` umieszczone u góry witryny PHP, a następnie wybierając jedną z dostępnych wersji:

- **Pliki źródłowe** (ang. *source*): jeśli Czytelnik nie chce korzystać z systemu zarządzania pakietami systemu Linux bądź jeśli ma zamiar skompilować PHP ze źródeł w systemie Windows, to powinien wybrać archiwum odpowiedniego typu. Przygotowywanie środowiska PHP ze źródeł w systemie Windows nie jest rozwiązaniem zalecanym i nie zostało ono opisane w tej książce. Za wyjątkiem bardzo szczególnych sytuacji gotowa wersja binarna PHP przygotowana dla systemu Windows powinna spełnić wszystkie nasze potrzeby i wymagania. Wersje te są dostępne w archiwach zapisanych w formatach *bzip2* lub *gzip*. Warto pamiętać, że ich zawartość jest taka sama — różne formaty kompresji zastosowano wyłącznie dla wygody użytkownika.
- **Pakiet zip dla systemu Windows** (ang. *zip package*): jeśli Czytelnik planuje używać PHP wraz z serwerem Apache w systemie Windows, powinien wybrać tę wersję, gdyż to właśnie ona została opisana w dalszej części rozdziału.
- **Instalator dla systemu Windows** (ang. *Windows installer*): ta wersja udostępni wygodny program instalacyjny, który nie tylko instaluje i konfiguruje środowisko PHP, ale pozwala także na konfigurację serwerów IIS, Apache oraz Xitami. Choć można jej używać w przypadku korzystania z serwera Apache, to jednak mnie nie udało się doprowadzić instalacji do pomyślnego zakończenia, a program zasugerował pobranie binarnego pakietu przeznaczonego dla systemu Windows. Dodatkowo, jeśli Czytelnik ma zamiar skonfigurować PHP do współpracy z serwerem IIS, powinien zajrzeć do podrozdziału „Instalacja serwera IIS i PHP w systemie Windows”. Dzięki współpracy firm Microsoft i Zend Technologies Ltd. proces ten został znacznie usprawniony, co szczegółowo opisałem.

Po wybraniu odpowiedniej wersji witryna określi kopię serwera położoną najbliżej naszego miejsca pobytu. Aby rozpocząć pobieranie, wystarczy wybrać jeden z dostępnych serwerów.

- 
- **Wskazówka** Jeśli mamy zamiar poeksperymentować z najnowszą, jeszcze rozwijaną wersją PHP, to można ją pobrać (zarówno w formie źródeł, jak i plików binarnych) z witryny <http://snaps.php.net/>. Należy pamiętać, że niektóre z tych wersji nie są przeznaczone do zastosowania w środowiskach produkcyjnych.
- 

## Pobieranie dokumentacji

Zarówno serwer Apache, jak i język PHP udostępniają doskonałą, wręcz wzorcową dokumentację, szczegółowo opisującą praktycznie każdy ich aspekt. Można je przeglądać w internecie (odpowiednio na stronach <http://httpd.apache.org> oraz [www.php.net](http://www.php.net)) lub pobrać i czytać na lokalnym komputerze.

## Pobieranie dokumentacji serwera Apache

W skład każdej dystrybucji serwera Apache wchodzi najnowsza wersja jego dokumentacji, zapisana w formatach XML oraz HTML i dostępna w różnych wersjach językowych. Jest ona umieszczana w katalogu *docs* w katalogu instalacyjnym.

W razie konieczności uaktualnienia posiadanej wersji dokumentacji, pobrania jej w innym formacie (takim jak PDF lub CHM — format pomocy systemu Windows) lub przeglądnienia jej na internecie należy zajrzeć na stronę <http://httpd.apache.org/docs-project>.

## Pobieranie dokumentacji PHP

Dokumentacja języka PHP jest dostępna w ponad 20 językach i w wielu różnych formatach, takich jak pojedynczy plik HTML, grupa wielu stron HTML czy też plik CHM. Wszystkie te wersje są generowane na podstawie głównych plików zapisanych w formacie DocBook, które można pobrać z serwera CVS projektu PHP (na przykład po to, by wygenerować dokumentację w jeszcze innym formacie). Dokumentacja jest umieszczona w katalogu *manual* wewnątrz katalogu instalacyjnego PHP.

Jeśli Czytelnik będzie musiał zaktualizować lokalną wersję dokumentacji lub zechce pobrać ją w innym formacie, powinien przejść na stronę <http://www.php.net/docs.php> i kliknąć odpowiednie łącze.

## Instalacja Apache i PHP w systemie Linux

Ten podrozdział opisuje proces budowania serwera Apache oraz PHP ze źródeł w systemach Linux. Potrzebny będzie do tego odpowiedni kompilator ANSI-C oraz system budowania; oba te komponenty są dostępne w systemach zarządzania pakietami we wszystkich głównych dystrybucjach Linuksa. Oprócz tego PHP wymaga dodatkowo narzędzi Flex (<http://flex.sourceforge.net/>) oraz Bison (<http://www.gnu.org/software/bison/bison.html>), natomiast serwer Apache wymaga języka Perl (w wersji co najmniej 5.003). Poza tym przeprowadzenie całego procesu wymaga dostępu do konta superużytkownika (root) na serwerze docelowym.

Dla wygody przed rozpoczęciem całego procesu warto przenieść oba pakiety do wybranego, jednego katalogu, na przykład: *usr/src/*. Poniżej opisano czynności, jakie należy wykonać w ramach procesu instalacji:

1. Rozpakować archiwa zip i tar z kodami źródłowymi serwera Apache i PHP. W zamieszczonych poniżej przykładowych poleceniach litery X reprezentują numer najnowszej stabilnej wersji pobranej zgodnie z informacjami podanymi we wcześniejszej części rozdziału.

```
%>gunzip httpd-2_X_XX.tar.gz
%>tar xvf httpd-2_X_XX.tar
%>gunzip php-XX.tar.gz
%>tar xvf php-XX.tar
```

2. Skonfigurować i zbudować serwer Apache. W najprostszym przypadku warto zastosować opcję `--enable-so`, która nakazuje włączenie możliwości wczytywania wspólnych modułów.

```
%>cd httpd-2_X_XX
%>./configure --enable-so [other options]
%>make
```

3. Zainstalować Apache (tę operację trzeba będzie wykonać z konta superużytkownika):

```
%>make install
```

4. Skonfigurować, zbudować i zainstalować PHP (dodatkowe informacje, dotyczące modyfikowania domyślnych ustawień instalacji oraz dołączania dodatkowych rozszerzeń, można znaleźć w punkcie „Konfiguracja PHP w trakcie budowania w systemach Linux”). W poniższym przykładzie `APACHE_INSTALL_DIR` należy zastąpić ścieżką dostępu do katalogu instalacyjnego serwera Apache, np. `/usr/local/apache2`:

```
%>cd ../php-X_XX
%>./configure --with-apxs2=APACHE_INSTALL_DIR/bin/apxs [other options]
%>make
%>make install
```

5. PHP jest dostarczany wraz z plikiem konfiguracyjnym, kontrolującym wiele aspektów działania języka. Plik ten jest znany jako *php.ini*, jednak w pakiecie z kodami źródłowymi nosi on nazwę *php.ini-dist*. Należy go skopiować do odpowiedniego katalogu i zmienić jego nazwę na *php.ini*. Podrozdział „Konfiguracja PHP”, zamieszczony nieco dalej, zawiera szczegółowe informacje na temat przeznaczenia oraz zawartości tego pliku. Warto zauważyć, że można go umieścić w dowolnym miejscu, jeśli jednak wybierzemy lokalizację inną niż domyślna, to należy odpowiednio skonfigurować PHP, używając w tym celu opcji `--with-config-file-path`. Warto także pamiętać, że mamy do dyspozycji również inny plik konfiguracyjny — *php.ini-recommended*. Określa on wartości różnych niestandardowych ustawień, a jego przeznaczeniem jest lepsze zabezpieczenie i zoptymalizowanie konfiguracji PHP, choć może się zdarzyć, że nie będzie ona w pełni zgodna z niektórymi starszymi aplikacjami PHP. Warto zastanowić się nad zastosowaniem tego pliku konfiguracyjnego zamiast, wspomnianego wcześniej, *php.ini-dist*. Jeśli Czytelnik zdecyduje się na to, powinien wykonać następujące polecenie:

```
%> cp php.ini-recommended /usr/local/lib/php.ini
```

6. Otworzyć plik konfiguracyjny serwera Apache, noszący nazwę *httpd.conf*, i upewnić się, że są w nim umieszczone przedstawione poniżej wiersze kodu. (Plik ten znajduje się w katalogu `APACHE_INSTALL_DIR/conf/httpd.conf`). Gdyby ich nie było, to należy je dodać. Warto zapisać je w tych miejscach pliku konfiguracyjnego, gdzie są umieszczone inne dyrektywy `LoadModule` oraz `AddType`:

```
LoadModule php5_module modules/libphp5.so
AddType application/x-httpd-php .php
```

Można mi wierzyć lub nie, ale to już wszystko. Teraz wystarczy ponownie uruchomić serwer Apache, wydając w tym celu polecenie:

```
%>/usr/local/apache/bin/apachectl restart
```

A teraz proszę kontynuować lekturę od rozdziału „Testowanie instalacji”.

- 
- **Wskazówka** Zastosowana w punkcie 6. dyrektywa `AddType` kojarzy konkretny typ MIME z konkretnym rozszerzeniem lub ich grupą. Rozszerzenie *.php* jest jedynie sugestią, zamiast niego można zastosować dowolne inne, na przykład *.html*, *.php5* lub *.json*. Oprócz tego z danym typem można skojarzyć więcej rozszerzeń, wystarczy je wszystkie zapisać w jednym wierszu, oddzielając od siebie znakami odstępu. Choć niektórzy preferują stosowanie PHP wraz z rozszerzeniem *.html*, to należy pamiętać, że w razie zastosowania takiego rozwiązania każde odebrane przez serwer żądanie dotyczące pliku HTML spowoduje przetworzenie jego zawartości przez interpreter PHP. Choć niektórzy mogą uznać takie rozwiązanie za wygodne, to jednak trzeba pamiętać, że spowoduje ono obniżenie wydajności serwera.
- 

## Instalacja Apache i PHP w systemie Windows

Wcześniejsze wersje serwera Apache przeznaczone dla systemów Windows nie były w żaden sposób optymalizowane pod ich kątem. Jednak serwer Apache2 został całkowicie przepisany właśnie w celu wykorzystania specyficznych cech tej platformy. Nawet jeśli Czytelnik nie planuje docelowo uruchamiać swojej aplikacji w systemie Windows, to jednak serwer ten będzie stanowił doskonałą platformę testową dla tych wszystkich osób, które preferują system Windows. Oto czynności, jakie należy wykonać w celu zainstalowania serwera Apache i PHP w systemie Windows:

1. Uruchomić program instalacyjny serwera Apache, dwukrotnie klikając ikonę pliku *apache\_X.X.XX-win32-x86-no\_ssl.msi*. Litera X w tej nazwie reprezentują numer najnowszej stabilnej wersji serwera, pobranej zgodnie z informacjami zamieszczonymi we wcześniejszej części rozdziału.
2. Proces instalacji rozpocznie się od wyświetlenia ekranu powitalnego. Warto poświęcić chwilę na przeczytanie zamieszczonych na nim informacji, a następnie należy kliknąć przycisk *Next*.
3. W następnym oknie zostanie wyświetlona umowa licencyjna. Należy ją uważnie przeczytać. Jeśli zgadzamy się z jej postanowieniami, można kliknąć przycisk *Next*.
4. Kolejne okno zawiera różne informacje dotyczące serwera. Należy je przeczytać i kliknąć przycisk *Next*.
5. Następnie program poprosi o podanie różnych informacji związanych z działaniem serwera, takich jak: nazwa domeny sieciowej, nazwa serwera oraz adres poczty elektronicznej administratora serwera. Jeśli Czytelnik zna te informacje, to powinien je podać; w przeciwnym razie w pierwszych dwóch polach wystarczy wpisać *localhost*, a w trzecim podać dowolny adres. Informacje te można później zmienić w dowolnym momencie w pliku konfiguracyjnym *httpd.conf*. Dodatkowo będziemy musieli podjąć decyzję, czy serwer Apache ma działać jako usługa dostępna dla wszystkich użytkowników systemu Windows, czy tylko dla bieżącego. Jeśli chcemy, by Apache był uruchamiany automatycznie podczas startu systemu, to należy wybrać opcję usługi dostępnej dla wszystkich użytkowników. Po dokonaniu wyboru należy przejść dalej, klikając przycisk *Next*.
6. Wybrać typ instalacji: *Typical* (typowa) lub *Custom* (dowolna). Jeśli Czytelnik chce wybrać instalowane elementy serwera, to należy wybrać opcję *Custom* — w kolejnym oknie dialogowym będzie można na przykład zaznaczyć, by nie instalować dokumentacji serwera. Jeśli jednak nie chcemy wprowadzać żadnych zmian, wystarczy zaznaczyć opcję *Typical* i kliknąć przycisk *Next*.

7. Program poprosi o wskazanie docelowego folderu. Domyślnie jest nim folder `C:\Program Files\Apache Group`. Warto jednak zastanowić się nad zmianą jego nazwy na `C:\apache`. Cokolwiek Czytelnik wybierze, powinien pamiętać, że w tej książce zostanie opisana instalacja serwera w katalogu `C:\apache`.
8. Kliknąć przycisk *Install*, by dokończyć proces instalacji. W ten sposób instalacja serwera Apache została zakończona. Teraz kolej na PHP.
9. Rozpakować pakiet instalacyjny PHP i umieścić jego zawartość w katalogu `C:\php`. Katalog instalacyjny może mieć dowolną nazwę, przy czym warto zwrócić uwagę, by nie zawierała ona znaków odstępu. W książce opisałem instalację w katalogu `C:\php`.
10. Przejsz do katalogu `C:\apache\conf` i otworzyć plik `httpd.conf` w edytorze.
11. Dodać do pliku konfiguracyjnego trzy przedstawione poniżej wiersze. Warto umieścić je bezpośrednio pod blokiem dyrektyw `LoadModule`, umieszczonym w sekcji `Global Environment` (środowisko ogólne).

```
LoadModule php_module c:/php/php5apache2_2.dll
AddType application/x-httpd-php .php
PHPIniDir "c:\php"
```

- 
- **Wskazówka** Zastosowana w punkcie 11. dyrektywa `AddType` kojarzy konkretny typ MIME z konkretnym rozszerzeniem lub ich grupą. Rozszerzenie `.php` jest jedynie sugestią, zamiast niego można zastosować dowolne inne, na przykład `.html`, `.php5` lub `.json`. Oprócz tego z danym typem można skojarzyć więcej rozszerzeń, wystarczy je wszystkie zapisać w jednym wierszu, oddzielając od siebie znakami odstępu. Choć niektórzy preferują stosowanie PHP wraz z rozszerzeniem `.html`, to należy pamiętać, że w razie zastosowania takiego rozwiązania każde odebrane przez serwer żądanie dotyczące pliku HTML spowoduje przetworzenie jego zawartości przez interpreter PHP. Choć niektórzy mogą uznać takie rozwiązanie za wygodne, to jednak trzeba pamiętać, że spowoduje ono obniżenie wydajności serwera. Dlatego też zalecane jest skorzystanie z ogólnie przyjętej konwencji i zastosowanie rozszerzenia `.php`.
- 

12. Zmienić nazwę pliku `php.ini-dist` na `php.ini` i skopiować go do katalogu `C:\php` (w przypadku wersji 5.3.0 PHP zmieniono nazwy plików konfiguracyjnych, które noszą teraz odpowiednio nazwy `php.ini-development` oraz `php.ini-production`; a zatem w razie korzystania z tej wersji PHP należy wybrać jeden z tych plików i zmienić jego nazwę). Plik konfiguracyjny `php.ini` zawiera setki dyrektyw odpowiedzialnych za określanie i dostosowywanie działania PHP. Opis jego przeznaczenia oraz szczegółowe informacje na temat zawartości można znaleźć w dalszej części rozdziału, w podrozdziale „Konfiguracja PHP”. Warto także pamiętać, że mamy do dyspozycji również inny plik konfiguracyjny — `php.ini-recommended`. Określa on wartości różnych niestandardowych ustawień, a jego przeznaczeniem jest lepsze zabezpieczenie i zoptymalizowanie konfiguracji PHP, choć może się zdarzyć, że nie będzie ona w pełni zgodna z niektórymi starszymi aplikacjami PHP. Warto zastanowić się nad zastosowaniem tego pliku konfiguracyjnego zamiast, wspomnianego wcześniej, `php.ini-dist`.
13. Jeśli Czytelnik używa jednego z systemów Windows NT, Windows 2000, XP, Vista<sup>1</sup>, to należy wybrać opcję *Start/Panel sterowania/Wydajność i konserwacja/Narzędzia administracyjne/Usługi*. W razie używania systemu Windows 98 proszę zajrzeć od informacji podanych pod koniec kolejnego punktu.
14. Odszukać Apache na liście i upewnić się, że usługa jest uruchomiona. Jeśli nie jest, to należy ją zaznaczyć i kliknąć opcję *Uruchom usługę* wyświetloną w lewej kolumnie. Jeśli serwer jest już uruchomiony, to należy zaznaczyć go na liście i kliknąć łącze *Uruchom ponownie usługę*; w ten sposób zostaną uwzględnione zmiany wprowadzone w pliku konfiguracyjnym `httpd.conf`. Następnie należy kliknąć pozycję serwera Apache prawym przyciskiem myszy i wybrać opcję *Właściwości*. Należy się upewnić, że na liście *Typ uruchamiania* jest wybrana opcja *Automatyczny*. Jeśli Czytelnik wciąż jeszcze używa systemu Windows 95/98, to konieczne będzie samodzielne ponowne uruchomienie serwera przy użyciu skrótu umieszczonego w menu *Start*.

---

<sup>1</sup> W systemie Windows 7 należy wybrać opcję *Start/Panel sterowania/System i zabezpieczenia/Narzędzia administracyjne/Usługi* — *przyp. tłum.*

# Instalacja serwera IIS i PHP w systemie Windows

Microsoft Windows wciąż pozostaje preferowanym systemem operacyjnym, nawet wśród programistów wspierających i promujących oprogramowanie otwarte; w końcu, wzięwszy pod uwagę jego dominującą pozycję rynkową, całkiem sensowne jest to, że większość osób preferuje korzystanie z tego dobrze znanego środowiska. Jednak ze względów stabilności oraz wydajności zdecydowanie najlepszym rozwiązaniem wciąż pozostanie wdrażanie aplikacji napisanych w PHP na serwerze Apache działającym w systemie operacyjnym Linux.

Rozwiązanie to niesie jednak ze sobą pewien problem, jeśli programista lubi pisać, a nawet wdrażać aplikacje na serwerze IIS działającym w systemie Windows. W ostatnich latach Microsoft, we współpracy z firmą Zend Technologies Ltd., poczynił ogromne starania mające na celu poprawienie zarówno stabilności, jak i wydajności środowiska PHP uruchamianego na IIS i Windows.

W 2009 roku Microsoft wykonał kolejny znaczący krok w kierunku bezproblemowego uruchamiania środowiska PHP na serwerze IIS, oddając do dyspozycji programistów produkt Microsoft Web Platform Installer. Produkt ten znacznie ułatwia instalację środowiska do tworzenia aplikacji internetowych w różnych konfiguracjach, także składających się z serwera IIS i PHP. Aby zainstalować IIS i PHP w systemie Windows 7, Vista, Server 2003 lub Server 2008, wystarczy wejść na witrynę <http://php.iis.net/> i kliknąć ogromny przycisk *Install PHP*.

Zakładając, że Czytelnik jeszcze nie zainstalował produktu Web Platform Installer, to niebawem powinien to zrobić. Zazwyczaj uruchomienie tego instalatora wymaga posiadania uprawnień administracyjnych. Po pobraniu programu Czytelnik zostanie poproszony o zainstalowanie PHP. W tej książce zostanie zastosowana nieco starsza wersja PHP (5.2.14), niemniej powinna ona w zupełności wystarczyć do wykonania przeważającej większości zamieszczonych przykładów. Aby kontynuować proces instalacji, proszę kliknąć przycisk *Install*, a następnie przeczytać i zaakceptować licencję. Można mi wierzyć lub nie, jednak po zakończeniu instalacji na używanym komputerze znajdzie się prawidłowo skonfigurowane środowisko PHP. Sposób jego przetestowania został opisany w kolejnym podrozdziale.

- 
- **Wskazówka** Microsoft Web Platform Installer nie działa w systemie Windows XP, nie oznacza to jednak, że osoby korzystające z tego systemu mają pecha. W lipcu 2010 roku Microsoft udostępnił darmowy produkt o nazwie IIS Developer Express (<http://learn.iis.net/>), który obsługuje system Windows XP oraz wszystkie najnowsze moduły serwera IIS 7, w tym także FastCGI, konieczny do uruchomienia PHP. Konfiguracja środowiska PHP i IIS jest nieco bardziej złożona w przypadku uruchamiania PHP i nie opiszę jej w tej książce, niemniej bardzo wiele informacji na ten temat można znaleźć w internecie.
- 

W czasie gdy pisałem tę książkę, konsola Web Platform Installer nie potrafiła zainstalować języka PHP, co oznacza, że należało instalować go ręcznie, korzystając z narzędzia do zarządzania programami systemu Windows. W systemie Windows 7 można je było uruchomić, klikając opcję *Odinstaluj program* w panelu sterowania.

## Testowanie instalacji

Najlepszym sposobem na sprawdzenie instalacji PHP jest próba wykonania skryptu napisanego w tym języku. A zatem należy uruchomić edytor tekstów, utworzyć nowy plik i wpisać w nim poniższe wiersze kodu:

```
<?php
    phpinfo();
?>
```

Proszę zapisać plik, nadając mu nazwę *phpinfo.php*. W przypadku korzystania z serwera Apache plik ten należy umieścić w katalogu *htdocs*. W razie korzystania z serwera IIS — w katalogu *C:\inetpub\wwwroot*.



Teraz pozostaje tylko otworzyć przeglądarkę i wyświetlić plik, podając adres `http://localhost/phpinfo.php`. Należy zwrócić uwagę, że nie można otworzyć tego skryptu, korzystając z opcji *Plik/Otwórz* przeglądarki, gdyż w takim przypadku nie zostanie on przetworzony i wykonany przez serwer.

Jeśli wszystko pójdzie dobrze, to w przeglądarce powinny zostać wyświetlone wyniki podobne do tych z rysunku 2.1. Jeśli Czytelnik próbuje uruchomić skrypt w witrynie działającej na serwerze WWW jakiegoś dostawcy usług internetowych i w efekcie uzyskuje komunikat informujący, że funkcja `phpinfo()` została wyłączona z powodów bezpieczeństwa, to konieczne będzie stworzenie innego skryptu testowego. W takim przypadku można skorzystać ze skryptu przedstawionego poniżej, którego wykonanie spowoduje wyświetlenie prostego komunikatu.

```
<?php
    echo "Prosty, lecz efektywny test działania PHP!";
?>
```

- 
- **Wskazówka** Wykonanie funkcji `phpinfo()` jest doskonałym sposobem zdobycia informacji na temat zainstalowanego środowiska PHP, gdyż przedstawia ona wiele danych na temat serwera WWW, jego środowiska oraz dostępnych rozszerzeń.
- 

PHP Version 5.3.0	
System	Windows NT PIOTREK 6.1 build 7600 ((null)) i586
Build Date	Jul 2 2009 21:02:53
Compiler	MSVC6 (Visual C++ 6.0)
Architecture	x86
Configure Command	cscrip/nologo configure.js
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	D:\Programming\kampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20090626
PHP Extension	20090626
Zend Extension	220090626
Zend Extension Build	API220090626,TS,VC6
PHP Extension Build	API20090626,TS,VC6
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
Zend Multibyte Support	disabled
IPv6 Support	enabled
Registered PHP Streams	https, ftps, php, file, glob, data, http, ftp, compress.zlib, compress.bzip2, phar, zip
Registered Stream Socket Transports	tcp, udp, ssl, sslv3, sslv2, tls
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*, consumed, dechunk, zlib.*, bzip2.*

*Rysunek 2.1. Wyniki generowane przez funkcję `phpinfo()`*

Jeśli podczas instalowania środowiska nie pojawiły się żadne zauważalne błędy, a pomimo to nie są wyświetlane oczekiwane wyniki, może to wynikać z jednego lub kilku poniższych problemów:

- Jeśli serwer Apache był konfigurowany własnoręcznie, to zmiany w jego plikach konfiguracyjnych nie zostaną uwzględnione aż do momentu jego powtórnego uruchomienia. Dlatego też należy pamiętać, by po wprowadzeniu zmian związanych z PHP do pliku konfiguracyjnego *httpd.conf* ponownie uruchomić serwer.
- Zastosowanie nieprawidłowych znaków lub poleceń w pliku konfiguracyjnym spowoduje, że nie uda się ponownie uruchomić serwera.
- Należy sprawdzić, czy każdy plik zawierający kod PHP posiada odpowiednie rozszerzenie, zdefiniowane w pliku konfiguracyjnym *httpd.conf*. Jeśli na przykład w pliku konfiguracyjnym określiliśmy, że tylko rozszerzenia *.php* będą reprezentowały pliki PHP, to nie należy próbować umieszczać kodu PHP w plikach z rozszerzeniem *.html*.
- Należy upewnić się, że kod PHP umieszczony w pliku został zapisany pomiędzy znacznikami `<?php` oraz `?>`. Jeśli tego nie zrobimy, to kod PHP może zostać wyświetlony w przeglądarce.
- Czytelnik utworzył plik o nazwie *index.php* i bezskutecznie stara się odwołać do niego tak jak do domyślnego pliku indeksu (poprzez podanie w adresie URL samej nazwy katalogu, bez nazwy pliku, na przykład: *www.przyklad.com/onas/* oraz *www.przyklad.com/onas/index.html*). Jednak serwer Apache uznaje za domyślne pliki indeksu wyłącznie pliki *index.html*, dlatego też konieczne będzie dodanie do pliku konfiguracyjnego Apache dyrektywy `DirectoryIndex` zawierającej nazwę naszego indeksu — *index.php*.

## Konfiguracja PHP

Choć podstawowa instalacja PHP w zupełności wystarczy dla większości początkujących użytkowników, to jednak istnieje duże prawdopodobieństwo, że już niedługo Czytelnik będzie chciał zmodyfikować domyślne ustawienia konfiguracyjne i poeksperymentować z dodatkami firm niezależnych, które nie są standardowo dołączane do PHP. W tym podrozdziale Czytelnik dowie się, jak można modyfikować działanie PHP i dostosowywać je do swoich szczególnych potrzeb.

## Konfiguracja PHP w trakcie tworzenia w systemach Linux

Zbudowanie PHP ze źródeł, opisane we wcześniejszej części rozdziału, w zupełności wystarczy, by rozpocząć pracę z tym językiem. Należy jednak pamiętać, że mamy do dyspozycji także wiele dodatkowych opcji. Pełną listę flag konfiguracyjnych (a jest ich ponad 200) można wyświetlić przy użyciu polecenia:

```
%>./configure --help
```

Aby zmodyfikować proces budowania PHP, wystarczy dodać jedną lub większą liczbę flag do polecenia `configure`, podając w razie konieczności odpowiednią wartość. Załóżmy na przykład, że chcielibyśmy wzbogacić tworzone środowisko PHP o możliwości korzystania z protokołu FTP, które domyślnie nie są włączone. W tym celu etap konfiguracji PHP należy zmodyfikować w następujący sposób:

```
%>./configure --with-apxs2=/usr/local/apache/bin/apxs --enable-ftp
```

W ramach kolejnego przykładu załóżmy, że chcemy korzystać z rozszerzenia *bzip2*. W tym celu wystarczy zmienić konfigurację PHP w poniższy sposób:

```
%>./configure --with-apxs2=/usr/local/apache/bin/apxs \  
>--with-bz2=[KATALOG_INSTALACYJNY]
```

Jednym z problemów, jakie najczęściej przytrafiają się początkującym użytkownikom, jest założenie, że zastosowanie odpowiedniej flagi konfiguracyjnej jest równoznaczne z udostępnieniem pożądaných możliwości w budowanym środowisku PHP. To nieprawda. Trzeba bowiem pamiętać, że konieczne jest także zainstalowanie oprogramowania, które w rzeczywistości odpowiada za obsługę danego rozszerzenia PHP. W ostatnim przykładzie, wykorzystującym rozszerzenie *bzip2*, konieczne będzie wcześniejsze zainstalowanie Java Development Kit (JDK).

## Modyfikowanie PHP budowanego w systemie Windows

Do wersji PHP 5.1 oraz 5.2.X (najnowszych wersji PHP dostępnych w czasie, gdy powstawała ta książka) dołączonych jest w sumie 45 rozszerzeń. Jednak użycie któregokolwiek z nich wymaga usunięcia znaku komentarza z odpowiedniego wiersza pliku konfiguracyjnego *php.ini*. Aby na przykład uaktywnić rozszerzenie XML-RPC, należy wprowadzić w pliku konfiguracyjnym kilka niewielkich modyfikacji:

1. Otworzyć plik *php.ini*, odszukać w nim dyrektywę `extension_dir` i przypisać jej wartość `C:\php\ext`. Jeśli Czytelnik zainstalował PHP w innym katalogu, to należy podać odpowiednią ścieżkę.
2. Odszukać wiersz `;extension=php_xmlrpc.dll`, usunąć znak średnika (;) z początku wiersza, zapisać modyfikację i zamknąć plik.
3. Ponownie uruchomić serwer WWW, co sprawi, że rozszerzenie stanie się dostępne i będzie można z niego korzystać w skryptach PHP. Warto także pamiętać, że niektóre rozszerzenia posiadają dodatkowe dyrektywy konfiguracyjne, które można znaleźć w dalszej części pliku konfiguracyjnego *php.ini*.

Czasami może się okazać, że włączanie pewnych rozszerzeń PHP będzie wymagało zainstalowania dodatkowego oprogramowania. Szczegółowe informacje dotyczące poszczególnych rozszerzeń można znaleźć w dokumentacji PHP.

## Konfiguracja PHP w trakcie jego działania

Zarówno w systemach Linux, jak i Windows istnieje możliwość zmiany zachowania środowiska PHP w trakcie jego działania. Zapewniają ją pliki konfiguracyjne *php.ini*. Zawiera on setki dyrektyw konfiguracyjnych kontrolujących działanie PHP oraz jego rozszerzeń. W dalszej części tego podrozdziału zostały przedstawione najczęściej stosowane dyrektywy konfiguracyjne PHP: ich przeznaczenie, zasięg, zakres oraz wartości domyślne.

### Korzystanie z dyrektyw konfiguracyjnych PHP

Zanim przejdziemy do poznawania poszczególnych dyrektyw konfiguracyjnych, w tym punkcie zostaną opisane różne sposoby manipulowania nimi — przy wykorzystaniu pliku konfiguracyjnego *php.ini*, plików konfiguracyjnych serwera Apache *httpd.conf* i *.htaccess* oraz bezpośrednio z poziomu skryptów PHP.

### Plik konfiguracyjny *php.ini*

W skład dystrybucji języka PHP wchodzi dwa wzorcowe pliki konfiguracyjne: *php.ini-dist* oraz *php.ini-recommended* (w wersji 5.3.0 języka noszą one odpowiednio nazwy: *php.ini-development* oraz *php.ini-production*). Instalując PHP, należy wybrać jeden z nich i zmienić jego nazwę na *php.ini* (dodatkowo w przypadku instalowania PHP w systemie Windows należy skopiować plik *php.ini* do katalogu wskazanego w dyrektywie `PHPIniDir` umieszczonej w pliku konfiguracyjnym *httpd.conf* serwera Apache). Zalecane jest zastosowanie drugiego z nich, gdyż przypisuje on sugerowane wartości wielu dyrektywom konfiguracyjnym. Zastosowanie się do tego zalecenia może zaoszczędzić wiele czasu i wysiłku, które w przeciwnym razie trzeba by poświęcić na zabezpieczanie i dostrajanie działania PHP — pamiętajmy bowiem, że plik *php.ini* zawiera ponad 200 dyrektyw konfiguracyjnych.

Choć domyślne wartości ustawień konfiguracyjnych w ogromnym stopniu pomagają w szybkim i bezproblemowym uruchomieniu PHP, to jednak najprawdopodobniej Czytelnik będzie chciał wprowadzić pewne zmiany w sposobie jego działania. Dlatego też warto dowiedzieć się czegoś więcej zarówno o nim, jak i o umieszczonych w nim parametrach konfiguracyjnych. Wyczerpujące informacje opisujące wiele dostępnych parametrów, ich przeznaczenie oraz zasięg można znaleźć w dalszej części rozdziału, w punkcie „Dyrektywy konfiguracyjne PHP”.

*Php.ini* jest globalnym plikiem konfiguracyjnym PHP, podobnie jak plik *httpd.conf* w przypadku serwera Apache. W wersji PHP 5.3.0 dokonano w nim znaczących zmian, jednak zarówno w wersjach wcześniejszych, jak i późniejszych jego zawartość została podzielona na 12 głównych sekcji:

- opcje języka,
- Safe mode (bezpieczny tryb działania PHP),
- kolorowanie składni,
- różne ustawienia,
- ograniczenia dostępu do zasobów,
- obsługa i rejestracja błędów (zagadnienia te zostały opisane w rozdziale 8.),
- obsługa danych,
- ścieżki dostępu i katalogi,
- przesyłanie plików na serwer (opisane w rozdziale 15.),
- opakowania strumieni,
- rozszerzenia wczytywane dynamicznie,
- ustawienia modułów.

Wiele spośród dyrektyw stosowanych w pliku *php.ini* opisałem w punkcie „Dyrektywy konfiguracyjne PHP”, zamieszczonym w dalszej części rozdziału. W kolejnych rozdziałach będą prezentowane ustawienia odnoszące się do opisywanych zagadnień.

Na początku warto poświęcić nieco czasu, by przyjrzeć się podstawowym, ogólnym cechom składni stosowanej w pliku *php.ini*. Jest to prosty plik tekstowy, zawierający wyłącznie komentarze oraz dyrektywy wraz z przypisywanymi im wartościami. Oto jego fragment:

```
;
; Allow the <? tag
;
short_open_tag = Off
```

Wiersze rozpoczynające się znakiem średnika (;) są komentarzami; w powyższym przykładzie parametrowi `short_open_tag` zostaje przypisana wartość `Off`.

Dokładny moment, gdy zmiany wprowadzane w pliku *php.ini* zostaną uwzględnione, zależy od sposobu zainstalowania PHP. Jeśli język został zainstalowany jako binarny plik CGI, plik konfiguracyjny jest odczytywany podczas każdego wywołania PHP, a to oznacza, że zmiany zostaną uwzględnione natychmiast. Jeśli jednak PHP został zainstalowany jako moduł serwera Apache, to plik konfiguracyjny *php.ini* jest odczytywany tylko raz — w momencie uruchamiania usługi (nazywanej także **demonem**) serwera. W tym przypadku, by zmiany zostały uwzględnione, konieczne jest ponowne uruchomienie serwera.

## Pliki `httpd.conf` oraz `.htaccess` serwera Apache

W przypadku gdy język PHP jest uruchamiany jako moduł serwera Apache, wiele spośród dyrektyw konfiguracyjnych PHP można modyfikować w plikach *httpd.conf* oraz *.htaccess*. Można to zrobić, poprzedzając określenie wartości dyrektywy konfiguracyjnej jednym z czterech słów kluczowych:

- `php_value`: ustawia wartość podanej dyrektywy.
- `php_flag`: ustawia wartość podanej dyrektywy logicznej.
- `php_admin_value`: ustawia wartość określonej dyrektywy. Różnica pomiędzy tym słowem kluczowym a `php_value` polega na tym, że `php_admin_value` nie można używać w plikach *.htaccess*, a wartości ustawionej przy jego użyciu nie można przesyłać w definicjach wirtualnych hostów ani w plikach *.htaccess*.
- `php_admin_flag`: ustawia wartość podanej dyrektywy. Różnica pomiędzy tym słowem kluczowym a `php_flag` polega na tym, że `php_admin_flag` nie można używać w plikach *.htaccess*, a wartości ustawionej przy jego użyciu nie można przesyłać w definicjach wirtualnych hostów ani w plikach *.htaccess*.

Abymy na przykład wyłączyć możliwość stosowania tak zwanych krótkich znaczników (ang. *short tags*) i jednocześnie uniemożliwić późniejsze modyfikacje tej dyrektywy, można umieścić w pliku *httpd.conf* następujący wiersz:

```
php_admin_flag short_open_tag Off
```

## Wewnątrz wykonywanego skryptu PHP

Trzecim sposobem określania ustawień konfiguracyjnych PHP, posiadającym najmniejszy zasięg oddziaływania, jest wykorzystanie funkcji `ini_set()`. Załóżmy na przykład, że chcemy zmodyfikować maksymalny dopuszczalny czas wykonywania skryptu. W tym celu wystarczy umieścić na samym początku pliku PHP następujący wiersz kodu:

```
ini_set('max_execution_time', '60');
```

## Zasięg dyrektyw konfiguracyjnych

Czy dyrektywy konfiguracyjne można modyfikować w dowolnym miejscu? Odpowiedź na to pytanie jest przecząca, a brak takiej możliwości jest w głównej mierze podyktowany względami bezpieczeństwa. Każda z dyrektyw konfiguracyjnych PHP ma swój zasięg i można ją modyfikować wyłącznie wewnątrz niego. Dostępne są cztery zasięgi dyrektyw:

- `PHP_INI_PERDIR`: wartość dyrektywy można określać w plikach *php.ini*, *httpd.conf* oraz *.htaccess*;
- `PHP_INI_SYSTEM`: wartość dyrektywy można określać w plikach *php.ini* oraz *httpd.conf*;
- `PHP_INI_USER`: wartość dyrektywy można określać w skryptach PHP;
- `PHP_INI_ALL`: wartość dyrektywy można określać w dowolnym miejscu.

## Dyrektywy konfiguracyjne PHP

W tej części rozdziału zostanie przedstawionych wiele podstawowych dyrektyw konfiguracyjnych PHP. Prócz ogólnej definicji dla każdej z nich zostanie podany jej zasięg oraz domyślna wartość. Ponieważ można przypuszczać, że Czytelnik będzie określał wartości dyrektyw w pliku *php.ini*, będą przedstawiał je w taki sposób, w jaki są zapisane w tym pliku.

Należy także pamiętać, że przedstawione tu dyrektywy w przeważającej większości odnoszą się do ogólnego działania PHP; dyrektywy związane z poszczególnymi rozszerzeniami lub z wybranymi aspektami PHP, szczegółowo opisywanymi w kolejnych rozdziałach książki, zostaną przedstawione w dalszych rozdziałach, w ramach prezentacji konkretnych zagadnień.

## Opcje języka

Dyrektywy umieszczone w tej części pliku konfiguracyjnego mają wpływ na najbardziej podstawowe aspekty działania PHP. Czytelnik na pewno będzie chciał poświęcić trochę czasu na ich dokładniejsze poznanie. Trzeba przy tym pamiętać, że zostały tu opisane jedynie wybrane, najczęściej używane dyrektywy. Warto poświęcić trochę czasu na przejrzanie pliku *php.ini* i sprawdzenie, jakie dyrektywy są dostępne.

- 
- **Ostrzeżenie** Choć dokumentacja PHP wciąż podaje domyślne wartości poszczególnych dyrektyw, to jednak reorganizacja i wydzielenie dwóch wersji pliku *php.ini*: *php.ini-development*, przeznaczonej do tworzenia aplikacji, oraz *php.ini-production*, przeznaczonej do użycia w środowiskach produkcyjnych — sprawia, że znaczenie słowa „domyślna” zostało uzależnione od kontekstu. Innymi słowy, wartości wielu dyrektyw użytych w wybranej wersji pliku *php.ini* będą inne od wartości tych samych dyrektyw stosowanych w drugiej wersji pliku. Z tego względu w tej książce postanowiłem postąpić wbrew przyjętym konwencjom i za wartości domyślne uznałem te, które zostały zastosowane w pliku konfiguracyjnym *php.ini-development*.
-

**engine = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: On.

Ten parametr określa, czy mechanizm PHP jest włączony, czy nie. Wyłączenie go spowoduje, że środowisko PHP w ogóle nie będzie używane. Oczywiście, jeśli planujemy używać PHP, powinniśmy pozostawić jego domyślną wartość.

**zend.ze1\_compatibility\_mode = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

Około trzy lata po udostępnieniu wersji 5.0 PHP wciąż powszechnie używana jest poprzednia wersja języka — PHP 4.X. Jednym z powodów tak przedłużonego cyklu aktualizacji używanej wersji języka są bardzo duże różnice w możliwościach obiektowych, występujące pomiędzy PHP 4 i 5. Ta dyrektywa stara się przywrócić niektóre wcześniejsze sposoby działania języka, umożliwiając uruchamianie, bez żadnych modyfikacji, aplikacji przystosowanych do PHP 4 w środowisku PHP 5.

---

■ **Uwaga** Dyrektywa `zend.ze1_compatibility_mode` nigdy nie działała zgodnie z zamierzeniami twórców i w wersji PHP 5.3.0 została usunięta.

---

**short\_open\_tag = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

Elementy skryptu PHP są zapisywane pomiędzy specjalnymi znacznikami. Dostępne są cztery formaty tych znaczników, a najkrótszy z nich jest określany jako krótkie znaczniki otwierające (ang. *short open tags*) i ma następującą postać:

```
<?
    echo "prosta instrukcja PHP";
?>
```

Być może Czytelnik zwrócił uwagę, że ta sama składnia jest używana w języku XML, co w niektórych okolicznościach może być przyczyną problemów. Dlatego też zapewniono możliwość wyłączenia obsługi tego formatu znaczników PHP. Jeśli dyrektywie `short_open_tag` zostanie przypisana wartość `On`, stosowanie krótkich znaczników otwierających będzie dozwolone, w przeciwnym razie nie będzie można ich używać.

**asp\_tags = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

PHP obsługuje także znaczniki stosowane w technologii ASP. Wyglądają one w następujący sposób:

```
<%
    echo "prosta instrukcja PHP";
%>
```

Jeśli Czytelnik korzystał wcześniej z tej technologii i chciałby dalej stosować znaną mu składnię, to może to zrobić, przypisując tej dyrektywie wartość `On`.

**precision = liczba\_całkowita**

Zasięg: PHP\_INI\_ALL; wartość domyślna: 14.

PHP obsługuje wiele typów danych, w tym także liczby zmiennoprzecinkowe. Ten parametr określa, ile liczb znaczących będzie wyświetlanych podczas prezentowania wartości zmiennoprzecinkowych. Warto zwrócić uwagę, że w systemach Windows dyrektywie tej przypisywana jest domyślnie wartość 12, natomiast w systemach Linux — wartość 14.

## y2k\_compliance = On | Off

Zasięg: PHP\_INI\_ALL; wartość domyślna: On.

Któż mógłby nie pamiętać o przerażającym problemie roku 2000, o którym było tak głośno dekadę temu? W rozwiązanie tego problemu zaangażowano nadludzkie wysiłki i choć jest to bardzo prawdopodobne, to jednak wciąż mogą się pojawić osoby używające przestarzałych przeglądarek, w których problem roku 2000 wciąż może występować. Jeśli z jakichś przedziwnych powodów Czytelnik jest przekonany, że znacząca część użytkowników odwiedzających jego witrynę zalicza się do tej kategorii, to może wyłączyć ten parametr, w przeciwnym razie należy pozostawić jego wartość domyślną.

## output\_buffering = On | Off | liczba\_całkowita

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna 4096.

Każdy, kto ma choćby minimalne doświadczenie w pisaniu skryptów PHP, bez wątplenia będzie znał poniższe komunikaty<sup>2</sup>:

---

```
"Cannot add header information – headers already sent"
"Oops, php_set_cookie called after header has been sent"
```

---

Są one generowane, gdy skrypt próbuje zmodyfikować nagłówki odpowiedzi po ich wysłaniu do przeglądarki użytkownika. Najczęstszą sytuacją ta występuje, gdy użytkownik próbuje ustawić wartość ciasteczka (ang. *cookie*) po przesłaniu fragmentu generowanej strony do przeglądarki użytkownika, co nie jest możliwe, gdyż nagłówki (niewidoczne przez użytkownika, lecz wykorzystywane przez przeglądarkę) zawsze są wysyłane przed jakąkolwiek treścią strony. W PHP 4.0 pojawiło się rozwiązanie tego uciążliwego problemu — buforowanie danych wyjściowych (ang. *output buffering*). Gdy mechanizm buforowania jest włączony, PHP przesyła wszystkie wygenerowane dane za jednym razem, po zakończeniu wykonywania skryptu. W ten sposób nagłówki można modyfikować w dowolnym miejscu skryptu, ponieważ żaden fragment odpowiedzi nie został jeszcze wysłany do przeglądarki użytkownika. Włączenie tej dyrektywy poprzez przypisanie jej wartości On spowoduje, że będzie stosowany mechanizm buforowania danych. Dodatkowo istnieje możliwość ograniczenia wielkości bufora (co niejawnie powoduje włączenie buforowania) poprzez określenie w dyrektywie `output_buffering` maksymalnej liczby bajtów, jakie można w nim będzie zapisać.

Jeśli Czytelnik nie planuje korzystać z buforowania danych wyjściowych, powinien wyłączyć ten mechanizm, gdyż powoduje on nieznaczne pogorszenie wydajności. Najprostszym rozwiązaniem problemu nagłówek jest, oczywiście, tworzenie ich przed przesłaniem do przeglądarki jakiegokolwiek treści generowanej strony.

## output\_handler = łańcuch

Zasięg: PHP\_INI\_ALL; wartość domyślna: NULL.

Ta interesująca dyrektywa nakazuje PHP przekazanie całych danych wyjściowych do funkcji przed ich ostatecznym przesłaniem do przeglądarki użytkownika. Załóżmy na przykład, że przed przesłaniem danych do przeglądarki chcielibyśmy je skompresować — obsługę takich danych zapewniają wszystkie nowoczesne przeglądarki zgodne z protokołem HTTP/1.1. Funkcję obsługującą dane wyjściowe można określić w poniższy sposób:

```
output_handler = "ob_gzhandler"
```

`ob_gzhandler()` jest funkcją kompresującą wchodzącą w skład biblioteki PHP obsługującej generację danych wyjściowych. Trzeba pamiętać, że nie można jednocześnie przypisać dyrektywie `output_handler` wartości `ob_gzhandler` i włączyć dyrektywy `zlib.output_compression` (opisanej poniżej).

---

<sup>2</sup> Nie można dodać informacji o nagłówku — nagłówki już zostały wysłane — *przypr. tłum.*

**zlib.output\_compression = On | Off | liczba\_całkowita**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: Off.

Kompresja danych wyjściowych może zaoszczędzić zarówno czas, jak i przepustowość łączy. Ta możliwość protokołu HTTP/1.1 jest obsługiwana przez znaczną większość nowoczesnych przeglądarek i bezpiecznie można jej używać w większości aplikacji. Automatyczną kompresję danych wyjściowych można włączyć, przypisując dyrektywie konfiguracyjnej `zlib.output_compression` wartość `On`. Dodatkowo można zarówno włączyć kompresję danych, jak i określić (w bajtach) wielkość używanego bufora — w tym celu wystarczy dyrektywie `zlib.output_compression` przypisać wartość będącą liczbą całkowitą.

**zlib.output\_handler = łańcuch\_znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Ta dyrektywa określa bibliotekę kompresji, jakiej należy użyć, jeśli biblioteka *zlib* nie będzie dostępna.

**implicit\_flush = On | Off**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: Off.

Włączenie tej dyrektywy sprawia, że bufor wyjściowy będzie automatycznie opróżniany po każdym wywołaniu funkcji `print()` lub `echo()` oraz po zakończeniu każdego bloku kodu HTML. Możliwość ta może się przydać, w sytuacjach gdy wykonanie skryptu zajmuje serwerowi wyjątkowo dużo czasu, na przykład gdy wykonuje on jakieś złożone obliczenia. W takich przypadkach można skorzystać z tej możliwości, by aktualizować kody statusu przesyłane do użytkownika, zamiast zmuszać go do czekania na zakończenie całych obliczeń.

**unserialize\_callback\_func = liczba\_całkowita**

Zasięg: PHP\_INI\_ALL; wartość domyślna: 100.

Ta dyrektywa pozwala określić odpowiedź mechanizmu deserializacji, w przypadku gdy zostanie podjęta próba utworzenia instancji niezdefiniowanej klasy. Dla większości użytkowników dyrektywa ta nie ma większego znaczenia, gdyż w takiej sytuacji, jeśli tylko będzie używany odpowiedni poziom raportowania błędów, PHP wygeneruje odpowiednie ostrzeżenie.

**serialize\_precision = liczba\_całkowita**

Zasięg: PHP\_INI\_ALL; wartość domyślna: 100.

Dyrektywa ta określa liczbę cyfr, jakie będą zapisywane podczas serializacji wartości zmiennoprzecinkowej (zarówno pojedynczej, jak i podwójnej precyzji — odpowiednio: `float` i `double`). Ustawienie odpowiedniej wartości tej dyrektywy może zapewnić, że procesy serializacji i deserializacji nie będą powodowały utraty danych.

**allow\_call\_time\_pass\_reference = On | Off**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: Off.

Argumenty funkcji można przekazywać na dwa sposoby: przez wartości oraz przez referencje. Konkretny sposób, w jaki poszczególne argumenty zostaną przekazane do funkcji w momencie jej wywołania, można określić w jej definicji, co zresztą jest sugerowanym rozwiązaniem. Niemniej włączając dyrektywę `allow_call_time_pass_reference`, można wymusić, by wszystkie argumenty były zawsze przekazywane przez referencję.

Szczegółowy opis przekazywania argumentów przez wartość oraz referencję, wraz z implikacjami obu tych sposobów, został zamieszczony w rozdziale 4., poświęconym funkcjom.

## Tryb bezpieczny

Jeśli PHP jest wdrażany w środowisku, z którego korzysta wielu użytkowników, takim jak współdzielone serwery oferowane przez wielu dostawców usług internetowych, to warto zastanowić się nad ograniczeniem jego możliwości. Jak łatwo się domyślić, zezwolenie wszystkim użytkownikom na korzystanie ze wszystkich



funkcji PHP mogłyby doprowadzić do niewłaściwego wykorzystania lub zniszczenia plików bądź innych zasobów serwera. W ramach zabezpieczenia przed taką ewentualnością język PHP udostępnia tak zwany **tryb bezpieczny** (ang. *safe mode*).

- 
- **Uwaga** Ze względu na pomyłki związane z nazwą tego mechanizmu oraz sposobem jego działania, a także z niezamierzonymi konsekwencjami użycia identyfikatorów użytkowników tworzących i posiadających prawa właścicieli plików tryb bezpieczny został wycofany z wersji PHP 5.3.0. Osobiście zdecydowanie odradzam jego stosowanie.
- 

Włączenie trybu bezpiecznego spowoduje wyłączenie kilku funkcji i możliwości PHP, które mogłyby być niebezpieczne, gdyby zostały niewłaściwie użyte w lokalnym skrypcie. Do funkcji tych zaliczają się między innymi: `parse_ini_file()`, `chmod()`, `chown()`, `chgrp()`, `exec()`, `system()` oraz operator odwrotnego apostrofu. Włączenie trybu bezpiecznego gwarantuje także, że właściciel wykonywanego skryptu będzie odpowiadał właścicielowi wszystkich plików i katalogów, jakie są przez ten skrypt używane. Ograniczenie to może jednak mieć nieoczekiwane i uciążliwe efekty uboczne, gdyż pliki niejednokrotnie mogą być kopiowane na serwer lub tworzone przez użytkowników o innych identyfikatorach.

Co więcej, włączenie trybu bezpiecznego umożliwia aktywację kilku dodatkowych ograniczeń, sterowanych za pośrednictwem dyrektyw umieszczanych w pliku konfiguracyjnym *php.ini*. Zostały one opisane poniżej.

### **safe\_mode = On | Off**

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `Off`.

Włączenie tej dyrektywy spowoduje, że możliwości PHP zaczną podlegać ograniczeniom trybu bezpiecznego.

### **safe\_mode\_gid = On | Off**

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `Off`.

W przypadku działania w trybie bezpiecznym włączenie dyrektywy `safe_mode_gid` wymusi sprawdzanie GID (identyfikatora grupy) podczas otwierania plików. Jeśli tryb bezpieczny jest wyłączony, stosowany jest bardziej restrykcyjny test polegający na sprawdzaniu identyfikatora użytkownika (UID).

### **safe\_mode\_include\_dir = łańcuch znaków**

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `NULL`.

Dyrektywa ta pozwala stworzyć bezpieczny azyl, który nie podlega ograniczeniom związanym ze sprawdzaniem identyfikatorów UID i GID, wykonywanym, gdy jest włączony tryb bezpieczny. W przypadku otwierania plików ze wskazanych katalogów testy UID i GID nie są wykonywane.

### **safe\_mode\_exec\_dir = łańcuch znaków**

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `NULL`.

W przypadku działania w trybie bezpiecznym dyrektywa `safe_mode_exec_dir` ogranicza możliwość wykonywania skryptów przy użyciu funkcji `exec()` wyłącznie do programów umieszczonych we wskazanym katalogu. Aby na przykład umożliwić wykonywanie jedynie funkcji dostępnych w katalogu `/usr/local/bin`, należy użyć poniższej dyrektywy:

```
safe_mode_exec_dir = "/usr/local/bin"
```

### **safe\_mode\_allowed\_env\_vars = łańcuch znaków**

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `PHP_*`.

Gdy włączony jest tryb bezpieczny, dyrektywa konfiguracyjna `safe_mode_allowed_env_vars` pozwala określić, które zmienne środowiskowe dotyczące systemu operacyjnego będzie można modyfikować z poziomu skryptów PHP. Na przykład zastosowanie poniższej dyrektywy sprawi, że będzie można modyfikować wyłącznie te zmienne, których nazwy rozpoczynają się od znaków `PHP_*`:

```
safe_mode_allowed_env_vars = "PHP_*
```

Trzeba pamiętać, że przypisanie tej dyrektywie pustego łańcucha znaków zapewni możliwość modyfikowania wszystkich zmiennych środowiskowych.

### **safe\_mode\_protected\_env\_vars = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: LD\_LIBRARY\_PATH.

Ta dyrektywa pozwala jawnie uniemożliwić modyfikowanie niektórych zmiennych środowiskowych. Aby na przykład nie dopuścić do modyfikacji zmiennych PATH oraz LD\_LIBRARY\_PATH, należy użyć następującej dyrektywy:

```
safe_mode_protected_env_vars = "PATH, LD_LIBRARY_PATH"
```

### **open\_basedir = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Podobnie jak dyrektywa DocumentRoot w przypadku serwera Apache, tak i dyrektywa open\_basedir PHP pozwala określić katalog bazowy, do którego będą się ograniczały wszelkie operacje na plikach. W ten sposób można uniemożliwić użytkownikom zapisywanie plików w chronionych obszarach serwera. Załóżmy na przykład, że wszelkie materiały związane z witryną WWW są umieszczone w katalogu /home/www. By uniemożliwić użytkownikom przeglądanie, a także potencjalną modyfikację takich plików jak /etc/passwd, wystarczy użyć dyrektywy open\_basedir o następującej postaci:

```
open_basedir = "/home/www/"
```

Trzeba przy tym pamiętać, że działanie tej dyrektywy nie jest zależne od tego, czy PHP pracuje w trybie bezpiecznym, czy nie.

### **disable\_functions = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

W pewnych okolicznościach Czytelnik może zdecydować, że trzeba całkowicie wyłączyć niektóre domyślne funkcje PHP, takie jak exec() bądź system(). Można to zrobić, wymieniając ich nazwy w dyrektywie konfiguracyjnej disable\_functions, jak pokazano na poniższym przykładzie:

```
disable_functions = "exec, system"
```

Trzeba pamiętać, że działanie tej dyrektywy nie jest zależne od tego, czy PHP pracuje w trybie bezpiecznym, czy nie.

### **disable\_classes = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Zważywszy na możliwości stosowania w PHP zasad programowania obiektowego, można sądzić, że już niebawem Czytelnik będzie stosował rozbudowane biblioteki klas. Jednak do tych bibliotek mogą należeć pewne klasy, które nie powinny być dostępne. Dzięki dyrektywie konfiguracyjnej disable\_classes uzyskujemy możliwość zablokowania wybranych klas. Aby na przykład uniemożliwić stosowanie dwóch klas — vector oraz graph — należałoby użyć następującej dyrektywy:

```
disable_classes = "vector, graph"
```

Trzeba przy tym pamiętać, że działanie tej dyrektywy nie jest zależne od tego, czy PHP pracuje w trybie bezpiecznym, czy nie.

### **ignore\_user\_abort = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

Jak wiele razy zdarzyło się nam wejść na jakąś stronę i wyjść z niej lub zamknąć przeglądarkę, zanim jej zawartość została w całości pobrana i wyświetlona? Takie postępowanie zazwyczaj nie ma żadnych konsekwencji. Ale co się stanie, jeśli serwer będzie w trakcie aktualizowania ważnych informacji w profilu

użytkownika lub kończenia transakcji handlowej? Włączenie dyrektywy konfiguracyjnej `ignore_user_abort` spowoduje, że serwer zignoruje żądanie przerwania sesji, zgłoszone przez użytkownika bądź przeglądarkę.

## Kolorowanie składni

PHP potrafi wyświetlać i modyfikować kolory, jakimi są prezentowane różne fragmenty kodu źródłowego. Tę możliwość można włączyć, zmieniając rozszerzenie skryptu na `.phps` (to domyślne rozszerzenie, które — jak Czytelnik się niebawem przekona — można zmienić) bądź korzystając z funkcji `show_source()` lub `highlight_file()`. Aby móc stosować rozszerzenie `.phps`, konieczne jest dodanie do pliku `httpd.conf` następującego wiersza:

```
AddType application/x-httpd-php-source .phps
```

Sześć przedstawionych poniżej dyrektyw konfiguracyjnych PHP pozwala określać kolor łańcuchów znaków, komentarzy, słów kluczowych, tła, domyślny kolor tekstu oraz komponentów HTML w wyświetlanym kodzie źródłowym. Każdej z tych dyrektyw można przypisać wartość RGB, szesnastkowe określenie koloru lub słowo kluczowe określające jego nazwę. Na przykład kolor czarny może być przedstawiony jako: `rgb(0,0,0)`, `#000000` lub `black`.

### **highlight.string = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `#DD0000`.

### **highlight.comment = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `#FF9900`.

### **highlight.keyword = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `#007700`.

### **highlight.bg = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `#FFFFFF`.

### **highlight.default = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `#0000BB`.

### **highlight.html = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `#000000`.

## Inne dyrektywy

Do tej kategorii zalicza się tylko jedna dyrektywa — `expose_php`.

### **expose\_php = On | Off**

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `On`.

Każdy strzępek informacji dotyczących naszego serwera WWW, jaki potencjalny napastnik może zdobyć, zwiększa szanse udanego włamania. Jednym z podstawowych sposobów zdobywania kluczowych informacji o charakterystykach serwera jest analiza jego podpisu. Na przykład serwer Apache domyślnie zwraca następujący tekst w nagłówkach każdej odpowiedzi:

```
Apache/2.2.0 (Unix) PHP/5.3.0 PHP/5.3.0-dev Server at www.example.com Port 80
```

Przypisanie dyrektywie `expose_php` wartości `Off` sprawi, że w podpisie (jeśli jego generowanie jest włączone) nie będą umieszczane informacje o zainstalowanym środowisku PHP. Choć odpowiednie zabezpieczenie serwera WWW wymaga także podjęcia innych kroków, to jednak takie ukrywanie informacji o nim jest ze wszech miar godne polecenia.

- 
- **Uwaga** Generowanie podpisu przez serwer Apache można wyłączyć, umieszczając w pliku konfiguracyjnym `httpd.conf` dyrektywę `ServerSignature` i przypisując jej wartość `Off`.
- 

## Ograniczenia zasobów

Choć w wersji PHP 5 poprawiono możliwości zarządzania zasobami, to jednak i tak trzeba zachować dużą ostrożność, by nie dopuścić do zmonopolizowania zasobów serwera przez aktualnie wykonywany skrypt w wyniku operacji zapoczątkowanych bądź to przez programistę, bądź użytkownika. Trzema podstawowymi obszarami, w których takie nadmierne zużycie zasobów systemowych przez skrypt jest najczęściej spotykane, są: czas wykonywania skryptu, czas przetwarzania przez skrypt danych wejściowych oraz zużycie pamięci. Każdy z nich można kontrolować przy użyciu jednej z trzech wymienionych poniżej dyrektyw.

### **max\_execution\_time = liczba\_całkowita**

Zasięg: `PHP_INI_ALL`; wartość domyślna: 30.

Parametr `max_execution_time` określa (w sekundach) górną granicę czasu wykonywania skryptu PHP. Przypisanie mu wartości 0 spowoduje, że nie będzie obowiązywał żaden limit czasu wykonywania. Warto pamiętać, że czas poświęcony na realizację jakichkolwiek programów zewnętrznych uruchomionych przez skrypt przy użyciu takich funkcji, jak `exec()` lub `system()`, nie jest wliczany do czasu jego wykonywania.

### **max\_input\_time = liczba\_całkowita**

Zasięg: `PHP_INI_ALL`; wartość domyślna: 60.

Ten parametr określa (w sekundach) maksymalny czas, jaki skrypt PHP może poświęcić na przetwarzanie danych przesłanych w żądaniu. Ma on szczególne znaczenie w przypadku przesyłania na serwer dużych plików, które są odbierane i przetwarzane przez skrypty PHP (zagadnienia te zostały opisane w rozdziale 15.).

### **memory\_limit = liczba\_całkowitaM**

Zasięg: `PHP_INI_ALL`; wartość domyślna: 128M.

Parametr `memory_limit` określa maksymalną ilość pamięci (wyrażoną w megabajtach), jaką można przydzielić skryptowi PHP.

## Obsługa danych

Parametry przedstawione w tym podpunkcie określają, w jaki sposób PHP będzie obsługiwać **zmienne zewnętrzne**, czyli zmienne przekazane do skryptu ze źródeł zewnętrznych. Potencjalnymi źródłami zmiennych zewnętrznych są żądania GET i POST, ciasteczka, system operacyjny oraz serwer WWW. Pozostałe opisane tu parametry określają domyślny zestaw znaków używany przez PHP i domyślny typ MIME oraz wskazują, czy do wyników generowanych przez skrypt będą dołączane jakieś pliki zewnętrzne.

### **arg\_separator.output = łańcuch\_znaków**

Zasięg: `PHP_INI_ALL`; wartość domyślna: `"&";`.

PHP potrafi automatycznie generować adresy URL, a do rozdzielania zmiennych wejściowych używa znaku `&`. Jeśli jednak konieczna będzie zmiana tej domyślnej konwencji, to można to zrobić, korzystając z dyrektywy konfiguracyjnej `arg_separator.output`.

### **arg\_separator.input = łańcuch\_znaków**

Zasięg: PHP\_INI\_ALL; wartość domyślna: ";&".

Standardowo do rozdzielania zmiennych wejściowych przekazywanych przy użyciu metod POST lub GET jest używany znak &. Gdyby się zdarzyło, choć jest to raczej bardzo mało prawdopodobne, że w tworzonej aplikacji należy zmodyfikować to domyślne ustawienie, można to zrobić przy użyciu dyrektywy `arg_separator.input`.

### **variables\_order = łańcuch\_znaków**

Zasięg: PHP\_INI\_ALL; wartość domyślna: "GPCS".

Ta dyrektywa określa kolejność, w jakiej będą przetwarzane zmienne `ENVIRONMENT`, `GET`, `POST`, `COOKIE` oraz `SERVER`. Jeśli dyrektywa `register_globals` jest włączona (choć pozornie obie dyrektywy nie są ze sobą powiązane), to kolejność podawanych wartości może doprowadzić do nieoczekiwanych rezultatów, gdyż zmienne analizowane później mogą przesłać wartości zmiennych przeanalizowanych wcześniej.

### **register\_globals = On | Off**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: Off.

Jeśli Czytelnik korzystał z PHP w wersji wcześniejszej niż 4.0, to samo wspomnienie o tej dyrektywie może wywołać zgrzytanie zębami i wrywanie włosów z głowy. W celu usunięcia związanych z nią problemów w wersjach PHP 4.2.0 i późniejszych dyrektywa ta została domyślnie wyłączona, co niestety zmusiło wielu użytkowników PHP o dłuższym stażu do całkowitego przemyślenia metodologii pisania aplikacji (a niejednokrotnie także ich przepisania). Wprowadzona zmiana służy najlepszym interesom programistów, znacząco wpływając na poprawę bezpieczeństwa aplikacji. Jeśli Czytelnik spotyka się z tymi zagadnieniami po raz pierwszy, to w ogóle nie ma o czym mówić.

We wczesnych wersjach PHP wszystkie zmienne zewnętrzne były automatycznie rejestrowane w globalnej przestrzeni nazw skryptu. Oznaczało to, że wszelkie zmienne zaliczane do kategorii `COOKIE`, `ENVIRONMENT`, `GET`, `POST` lub `SERVER` automatycznie stawały się dostępne globalnie. A ponieważ były globalnie dostępne, można je było także modyfikować bez żadnych ograniczeń. Choć można było sądzić, że jest to wygodne rozwiązanie, to jednak stanowiło ono potencjalne zagrożenie, gdyż zmienne, które powinny być zarządzane wyłącznie za pośrednictwem ciasteczek, można było także modyfikować za pośrednictwem adresu URL. Załóżmy na przykład, że identyfikator sesji, jednoznacznie określający użytkownika, był przekazywany pomiędzy poszczególnymi stronami w ciasteczku. Nikt oprócz konkretnego użytkownika nie powinien mieć dostępu do danych skojarzonych z użytkownikiem określanym przez dany identyfikator. Użytkownik mógł jednak wyświetlić wartość ciasteczka, skopiować ją i dodać do adresu URL, jak pokazano na poniższym przykładzie:

```
http://www.example.com/secretdata.php?sessionId=4x5bh5H793adK
```

Następnie użytkownik mógł przesłać ten adres w wiadomości poczty elektronicznej do innej osoby. Gdyby aplikacja nie używała żadnych dodatkowych zabezpieczeń (takich jak kontrola adresu IP), to ta druga osoba mogłaby uzyskać dostęp do poufnych danych. Wyłączenie dyrektywy `register_globals` zapobiega występowaniu takich zagrożeń. Choć zmienne zewnętrzne wciąż są dostępne w globalnej przestrzeni nazw, to do każdej z nich należy się odwoływać przy użyciu odpowiedniego typu. Na przykład do zmiennej `sessionId` zastosowanej w ostatnim przykładzie można się odwoływać wyłącznie w następujący sposób:

```
$_COOKIE['sessionId'];
```

Próba zmiany wartości tej zmiennej przy użyciu jakiegokolwiek innego odwołania (takiego jak `GET` lub `POST`) spowoduje utworzenie nowej zmiennej globalnej (takiej jak `$_GET['sessionId']` lub `$_POST['sessionId']`). Szczegółowe informacje dotyczące zmiennych zewnętrznych wszelkich typów (`COOKIE`, `ENVIRONMENT`, `GET`, `POST` oraz `SERVER`) zostały zamieszczone w rozdziale 3., w jego części poświęconej zmiennym superglobalnym.

Choć wyłączenie dyrektywy `register_globals` jest bezsprzecznie dobrym pomysłem, to jednak nie jest to jedyny czynnik, jaki należy brać pod uwagę w ramach zabezpieczania aplikacji. Znacznie więcej informacji dotyczących bezpieczeństwa i zabezpieczenia aplikacji PHP zostało zamieszczonych w rozdziale 21.

- **Uwaga** Przez wiele lat dyrektywa `register_globals` oraz łączące się z nią aspekty działania PHP były źródłem nieustających problemów związanych z bezpieczeństwem aplikacji PHP. Dlatego też, poczynając od PHP 5.3.0, jej stosowanie nie jest zalecane.

## register\_long\_arrays = On | Off

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `Off`.

Ta dyrektywa określa, czy wciąż należy rejestrować tablice wejściowe (`ENVIRONMENT`, `GET`, `POST`, `COOKIE` oraz `SERVER`), używając stosowanych niegdyś długich nazw — `HTTP_*_VARS`. Ze względów wydajności działania języka zaleca się wyłączenie tej dyrektywy.

- **Uwaga** W języku PHP 5.3.0 stosowanie tej dyrektywy nie jest zalecane.

## Stosowanie apostrofów i cudzysłowów

Już od wielu lat zarówno cudzysłowy, jak i apostrofy odgrywają w językach programowania szczególną rolę. Ponieważ są one powszechnie stosowane do wyróżniania fragmentów tekstów w kodach programów oraz w języku pisanim, zatem, by uniknąć problemów, konieczne będzie poznanie jakiegoś sposobu pozwalającego na ich rozróżnianie. Rozwiązanie jest całkiem proste: te znaki cudzysłowów bądź apostrofów, które nie ograniczają łańcucha znaków, należy poprzedzać odwrotnym ukośnikiem. Jeśli nie będziemy stosować tej zasady, to mogą się pojawić niespodziewane błędy. Rozważmy następujący przykład:

```
$sentence = "Janek powiedział: "Kocham samochody wyścigowe!";
```

Które z zastosowanych w nim znaków cudzysłowu ograniczają cały łańcuch znaków, a które wypowiedź Janka? Język PHP nie pozna odpowiedzi na to pytanie, jeśli niektóre z cudzysłowów nie zostaną odpowiednio oznaczone:

```
$sentence = "Janek powiedział: \"Kocham samochody wyścigowe!\"";
```

Poprzedzanie odwrotnym ukośnikiem znaków cudzysłowu lub apostrofu, które nie ograniczają łańcuchów znaków, określamy jako **włączanie magicznych cudzysłowów**. Można to zrobić automatycznie — włączając dyrektywę `magic_quotes_gpc` (opisaną w dalszej części rozdziału) — bądź też ręcznie, przy użyciu funkcji `addslashes()` oraz `stripslashes()`. Zalecane jest zastosowanie tej drugiej strategii, gdyż zapewnia ona programiście całkowitą kontrolę nad działaniem aplikacji (jednak w sytuacjach gdy próbujemy skorzystać z aplikacji oczekujących, że mechanizm automatycznego poprzedzania cudzysłowów i apostrofów będzie włączony, konieczne będzie jego uaktywnienie).

Ze względu na problemy, jakich mechanizm ten przysparza programistom, w PHP 5.3.0 nie zaleca się jego stosowania.

## register\_argc\_argv = On | Off

Zasięg: `PHP_INI_SYSTEM`; wartość domyślna: `Off`.

Przekazywanie zmiennych informacji przy wykorzystaniu metody `GET` przypomina przekazywanie argumentów do programu wykonywalnego. W wielu językach programowania informacje o argumentach przekazywanych do programów są dostępne pod postacią zmiennych `argc` oraz `argv`. Pierwsza z nich zawiera liczbę przekazanych argumentów, natomiast druga jest indeksowaną liczbami tablicą zawierającą same argumenty. Jeśli chcemy, by zmienne `$argc` i `$argv` były deklarowane, co pozwoli nam stosować ten sposób przetwarzania danych wejściowych, należy przypisać dyrektywie `register_argc_argv` wartość `On`.

**post\_max\_size = liczba\_całkowitaM**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: 8M.

Spośród dwóch dostępnych metod przekazywania danych w żądaniach HTTP metoda POST znacznie lepiej nadaje się do przesyłania dużych ilości danych, na przykład informacji podawanych w formularzach. Niemniej zarówno ze względów bezpieczeństwa, jak i wydajności działania można zdecydować się na ograniczenie maksymalnej wielkości danych, jakie będzie można przysyłać do skryptu PHP tą metodą. Właśnie do tego celu służy dyrektywa `post_max_size`.

**magic\_quotes\_gpc = On | Off**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: Off.

Ten parametr określa, czy magiczne cudzysłowy będą operować na danych przesyłanych przy wykorzystaniu GET, POST oraz w ciasteczkach. W razie jego uaktywnienia wszelkie znaki apostrofów, cudzysłowów, odwrotnych ukośników oraz znaki puste będą automatycznie poprzedzane znakiem odwrotnego ukośnika.

**magic\_quotes\_runtime = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

Włączenie tego parametru spowoduje automatyczne poprzedzanie znakiem odwrotnego ukośnika wszelkich apostrofów i cudzysłowów umieszczonych w danych pochodzących ze źródeł zewnętrznych, takich jak bazy danych lub pliki tekstowe.

**magic\_quotes\_sybase = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

Ten parametr ma znaczenie wyłącznie wtedy, gdy włączony jest także parametr `magic_quotes_runtime`. Jeśli parametr `magic_quotes_sybase` zostanie włączony, to wszystkie apostrofy i cudzysłowy umieszczone w danych pochodzących ze źródeł zewnętrznych, będą poprzedzane znakiem apostrofu, a nie odwrotnego ukośnika. Rozwiązanie to jest przydatne, gdy informacje są pobierane z bazy danych Sybase, która wykorzystuje ten raczej nietypowy sposób oznaczania znaków specjalnych.

**auto\_prepend\_file = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Tworzenie szablonów nagłówek i dołączanie bibliotek przed wykonaniem skryptów PHP najczęściej jest realizowane przy wykorzystaniu funkcji `include()` lub `require()`. Można jednak zautomatyzować ten proces i uniknąć konieczności stosowania tych funkcji w tworzonych skryptach poprzez podanie nazw dołączanych plików wraz z odpowiednimi ścieżkami dostępu w dyrektywie `auto_prepend_file`.

**auto\_append\_file = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Dołączanie stopek za wynikami generowanymi przez skrypty PHP najczęściej jest realizowane przy wykorzystaniu funkcji `include()` lub `require()`. Można jednak zautomatyzować ten proces i uniknąć konieczności stosowania tych funkcji w tworzonych skryptach poprzez podanie nazw dołączanych plików wraz z odpowiednimi ścieżkami dostępu w dyrektywie `auto_append_file`.

**default\_mimetype = łańcuch znaków**

Zasięg: PHP\_INI\_ALL; wartość domyślna: "text/html".

Typy MIME stanowią standardowy sposób klasyfikowania typów plików używanych w internecie. Aplikacje PHP zapewniają możliwości udostępniania plików dowolnych typów, jednak najbardziej popularnym z nich jest `text/html`. Jeśli jednak aplikacje PHP są używane w innych celach, na przykład jako generatory treści dla aplikacji WML (ang. *Wireless Markup Language*), konieczne będzie odpowiednie dostosowanie typu MIME generowanych odpowiedzi. Można to zrobić, podając go w dyrektywie `default_mimetype`.

## default\_charset = łańcuch znaków

Zasięg: PHP\_INI\_ALL; wartość domyślna: NULL.

W wersji 4.0 PHP informacje o sposobie kodowania są zwracane w nagłówku Content-Type. Domyślnie używany jest zestaw znaków iso-8859-1, obsługujący między innymi języki angielski, hiszpański, niemiecki oraz portugalski. Jeśli jednak w tworzonej aplikacji PHP są używane inne języki, takie jak japoński, chiński bądź hebrajski, to dyrektywa default\_charset pozwala podać odpowiedni zestaw znaków.

## always\_populate\_raw\_post\_data = On | Off

Zasięg: PHP\_INI\_PERDIR; wartość domyślna: Off.

Przypisanie tej dyrektywie wartości On powoduje, że PHP będzie zapisywać w zmiennej \$HTTP\_RAW\_POST\_DATA łańcuch znaków zawierający pary nazwa – wartość przesłane w żądaniu POST. Dotyczy to nawet tych sytuacji, gdy przekazana z formularza zmienna nie ma żadnej wartości. Załóżmy na przykład, że dyrektywa ta została włączona, a Czytelnik dysponuje formularzem zawierającym dwa pola tekstowe — pierwsze z nich pozwala na podanie imienia użytkownika, a drugie — jego adresu poczty elektronicznej. W odpowiedzi na przesłanie formularza wykonywany jest skrypt zawierający tylko jedną instrukcję:

```
echo $HTTP_RAW_POST_DATA;
```

W takim przypadku przesłanie pustego formularza spowoduje wygenerowanie następujących wyników:

---

```
name=&email=
```

---

Jeśli natomiast użytkownik wypełni oba pola i kliknie przycisk przesyłający go na serwer, to wyniki będą mieć następującą postać:

---

```
name=jason&email=jason%40example.com
```

---

## Ścieżki i katalogi

W tej części rozdziału zostały przedstawione dyrektywy określające domyślne ustawienia ścieżek. Ścieżki te są używane przez PHP do dołączania bibliotek oraz rozszerzeń do określania katalogów, w jakich są umieszczane pliki użytkownika, oraz głównych katalogów serwera.

### include\_path = łańcuch znaków

Zasięg: PHP\_INI\_ALL; wartość domyślna: NULL.

Ścieżka określona w tym parametrze jest używana przez takie funkcje, jak: include(), require() oraz fopen\_with\_path(), jako główna ścieżka. Można w nim podać więcej ścieżek, oddzielając je od siebie znakiem średnika, jak pokazano na poniższym przykładzie:

```
include_path=".:usr/local/include/php;/home/php"
```

Domyślnie temu parametrowi jest przypisywana wartość zmiennej środowiskowej o nazwie PHP\_INCLUDE\_PATH. Warto zwrócić uwagę, że w systemach Windows zamiast ukośnika w ścieżkach są stosowane znaki odwrotnego ukośnika, a na ich początku jest umieszczana litera określająca dysk:

```
include_path=".;C:\php\includes"
```

### doc\_root = łańcuch znaków

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Parametr ten określa domyślny katalog, z jakiego będą pobierane wszystkie skrypty PHP. Jest on używany wyłącznie wtedy, gdy jego wartość nie jest pusta.



**user\_dir = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: NULL.

Parametr `user_dir` określa bezwzględną ścieżkę dostępu do katalogu, wykorzystywanego przez PHP podczas otwierania plików, do których będziemy się odwoływać przy użyciu zapisu `/~uzytkownik`. Jeśli na przykład przypisana mu zostanie wartość `/home/users`, a spróbujemy się odwołać do pliku `~gilmore/collections/books.txt`, to PHP określi, że bezwzględna ścieżka dostępu do pliku będzie mieć postać: `/home/users/gilmore/collections/books.txt`.

**extension\_dir = łańcuch znaków**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: `./` (w systemie Windows domyślną wartością jest `ext`).

Dyrektywa `extension_dir` informuje PHP, gdzie są przechowywane rozszerzenia, które można dynamicznie wczytywać. Domyślnie przypisywana jest jej wartość `./`, oznaczająca, że rozszerzenia są umieszczone w tym samym katalogu co wykonywany skrypt. W środowisku Windows, jeśli wartość tej dyrektywy nie została określona, to domyślnie zostanie zastosowany katalog `C:\KATALOG-INSTALACYJNY-PHP\ext\`.

**enable\_dl = On | Off**

Zasięg: PHP\_INI\_SYSTEM; wartość domyślna: `Off`.

Funkcja `enable_dl()` pozwala użytkownikom na wczytywanie rozszerzeń w trakcie wykonywania skryptu.

**Opakowania używane przez funkcję fopen**

W tej części rozdziału zostało opisanych pięć dyrektyw konfiguracyjnych związanych z dostępem i operacjami na zdalnych plikach.

**allow\_url\_fopen = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: `On`.

Przypisanie tej dyrektywie wartości `On` spowoduje, że PHP będzie traktować zdalne pliki niemal w taki sam sposób jak pliki lokalne. W takim przypadku skrypt PHP może uzyskać dostęp i wykonywać operacje na plikach przechowywanych na zdalnych serwerach, jeśli tylko uprawnienia tych plików na to pozwalają.

**from = łańcuch znaków**

Zasięg: PHP\_INI\_ALL; wartość domyślna: NULL.

Nazwa tej dyrektywy jest prawdopodobnie dosyć myląca, gdyż nie określa ona tożsamości, lecz hasło anonimowego użytkownika, używane do nawiązywania połączeń FTP. Dlatego też, jeśli w pliku konfiguracyjnym określimy jej wartość w następujący sposób:

```
from = "jason@example.com"
```

to podczas uwierzytelniania użytkownika na serwer FTP zostaną przekazane nazwa `anonymous` oraz hasło `jason@example.com`.

**user\_agent = łańcuch znaków**

Zasięg: PHP\_INI\_ALL; wartość domyślna: NULL.

Wraz z danymi generowanymi przez skrypt PHP zawsze przesyła do klienta także nagłówki odpowiedzi, a wśród nich nagłówek określający przeglądarkę użytkownika. Jego zawartość można ustalić przy użyciu dyrektywy `user_agent`.

**default\_socket\_timeout = liczba\_całkowita**

Zasięg: PHP\_INI\_ALL; wartość domyślna: 60.

Ta dyrektywa określa czas oczekiwania (wyrażony w sekundach) stosowany w operacjach na strumieniach bazujących na gniazdach.

**auto\_detect\_line\_endings = On | Off**

Zasięg: PHP\_INI\_ALL; wartość domyślna: Off.

Jednym z niewyczerpanych źródeł frustracji programistów są problemy związane ze znakiem końca wiersza tekstu. Wynikają one z faktu, iż w różnych platformach systemowych znaki te mają różną postać. Włączenie dyrektywy `auto_detect_line_endings` określa, czy dane wczytywane przy użyciu funkcji `fgets()` oraz `file()` będą używały konwencji stosowanej na komputerach Macintosh, w systemie MS-DOS, czy też Linux.

## Rozszerzenia dynamiczne

Ta część rozdziału jest poświęcona tylko jednej dyrektywie — `extension`.

**extension = łańcuch\_znaków**

Zasięg: PHP\_INI\_ALL; wartość domyślna: NULL.

Dyrektywa `extension` służy do dynamicznego wczytywania konkretnego modułu. W systemach Windows rozszerzenie można wczytać, używając dyrektywy o następującej postaci:

```
extension = php_bz2.dll
```

W systemach Unix wygląda ona nieco inaczej:

```
extension = php_bz2.so
```

Trzeba pamiętać, że niezależnie od używanego systemu operacyjnego, samo zastosowanie tej dyrektywy nie zawsze spowoduje udostępnienie rozszerzenia. Konieczne będzie także zainstalowanie na komputerze odpowiedniego oprogramowania.

## Wybór edytora

Choć nic nie stoi na przeszkodzie, by rozpoczynać przygodę z pisaniem skryptów PHP, korzystając z najprostszych edytorów tekstów, takich jak Notatnik (w systemie Windows) bądź `vi` (w systemach Unix i Linux), to jednak istnieje duże prawdopodobieństwo, że stosunkowo szybko Czytelnik zechce skorzystać z rozbudowanego narzędzia, przeznaczonego do tworzenia aplikacji PHP. Dostępnych jest kilka takich narzędzi i to zarówno darmowych, jak i komercyjnych.

## Adobe Dreamweaver CS5

Program Dreamweaver CS5 firmy Adobe przez wiele osób jest uważany za najlepsze dostępne narzędzie do projektowania i tworzenia stron WWW. Program Dreamweaver CS3, stworzony jako kompletne narzędzie, obsługuje wszystkie kluczowe technologie, takie jak Ajax, CSS, HTML, JavaScript, PHP oraz XML, których połączenie pozwala tworzyć doskonałe i nowoczesne witryny WWW.

Dreamweaver CS5 nie tylko pozwala tworzyć strony HTML w środowisku WYSIWYG (ang. *what you see is what you get* — dostajesz to, co widzisz), lecz także udostępnia wiele przydatnych narzędzi, ułatwiających programistom PHP tworzenie kodu i zarządzanie nim, takich jak: kolorowanie składni, automatyczne uzupełnianie kodu, mechanizmy łatwego zapisywania i wielokrotnego stosowania fragmentów kodu.

Program Adobe Dreamweaver CS5 ([www.adobe.com/products/dreamweaver](http://www.adobe.com/products/dreamweaver)) jest dostępny w wersjach przeznaczonych dla systemów Windows oraz Mac OS X i kosztuje 399 dol.<sup>3</sup>

<sup>3</sup> Na polskim rynku program ten można kupić za około 1800 zł — *przyj. tłum.*

## Notepad++

Notepad++ jest dojrzałym programem otwartym, pomyślanym jako zamiennik prostego Notatnika w systemach Windows. Program ten, przetłumaczony na wiele języków, udostępnia mnóstwo użytecznych narzędzi, których nie powstydziliby się zaawansowane środowisko programistyczne (IDE), takich jak: tworzenie zakładek wskazujących na konkretne wiersze kodu, oznaczanie par odpowiadających sobie nawiasów klamrowych, kolorowanie składni, wygodna obsługa wcięć, rozbudowane funkcje wyszukiwania, rejestracja makr ułatwiających wykonywanie żmudnych czynności (jak choćby wstawianie komentarzy).

Wsparcie dla języka PHP nie jest szczególnie rozbudowane, a wygodę programowania zapewniają raczej ogólne niż dostosowane do PHP możliwości programu. Choć dostępne są podstawowe możliwości automatycznego uzupełniania nazw funkcji, które mogą nam nieco uprościć tworzenie kodu, to jednak — jeśli chodzi o nazwy parametrów oraz ich kolejność — jesteśmy pozostawieni sami sobie.

Notepad++ działa wyłącznie w systemie Windows. Twórcy rozpowszechniają go na zasadach licencji GNU GPL. Więcej informacji na jego temat można znaleźć na stronie <http://notepad-plus.sourceforge.net>.

## PDT (PHP Development Tools)

Projekt PDT ([www.eclipse.org/pdt](http://www.eclipse.org/pdt)) jest aktualnie bardzo aktywnie rozwijany. Projekt ten jest wspierany przez firmę Zend Technologies Ltd. ([www.zend.com](http://www.zend.com)) i wykorzystuje otwartą platformę Eclipse ([www.eclipse.org](http://www.eclipse.org)) — niezwykle popularne, rozszerzalne środowisko do tworzenia narzędzi programistycznych. Ma on szansę stać się liderem, jeśli chodzi o zintegrowane środowiska programistyczne (IDE) do tworzenia aplikacji PHP, przeznaczone zarówno dla hobbystów, jak i profesjonalistów.

- 
- **Uwaga** W oparciu o platformę Eclipse stworzono bardzo wiele narzędzi ułatwiających wykonywanie kluczowych zadań związanych z programowaniem, poczynając od modelowania danych, poprzez projektowanie logiki biznesowej, raportowanie, testowanie, sprawdzanie wydajności, a kończąc na tworzeniu kodu. Choć Eclipse najbardziej kojarzy się z IDE do tworzenia aplikacji pisanych w Javie, to jednak w oparciu o nie stworzono podobne IDE do pisania programów w C, C++, Cobolu, a ostatnio także w PHP.
- 

## Zend Studio

Zend Studio jest bezsprzecznie najpotężniejszym z aktualnie dostępnych narzędzi do pisania aplikacji PHP i to zarówno otwartych, jak i komercyjnych. Ten flagowy produkt firmy Zend Technologies Ltd. udostępnia wszystkie możliwości, jakich można by oczekiwać od zintegrowanego środowiska programistycznego klasy korporacyjnej, poczynając od doskonałego mechanizmu uzupełniania kodu, poprzez profilowanie kodu, na wygodnej obsłudze procesu wdrażania aplikacji kończąc.

Oprócz tego Zend Studio ułatwia tworzenie kodu integrującego aplikację z popularnymi bazami danych, takimi jak MySQL, Oracle, PostgreSQL oraz SQLite, i zapewnia możliwość wykonywania zapytań SQL, zarządzania schematami baz danych oraz ich zawartością.

Zend Studio ([www.zend.com/products/studio](http://www.zend.com/products/studio)) jest dostępne w wersjach przeznaczonych dla systemów Windows, Linux oraz Mac OS X za 299 dol.

## Wybór firmy udostępniającej serwery WWW

Jeśli Czytelnik nie jest zatrudniony lub nie współpracuje z firmą, która już dysponuje środowiskiem zawierającym serwer WWW, to wcześniej czy później konieczne będzie sprawdzenie dostępnych ofert i wybór jednej z firm udostępniających serwery. Na szczęście na tym rynku działa bardzo wiele firm i konkurencja jest ogromna. Dzięki temu nietrudno znaleźć oferty firm, które za bardzo niewielkie stawki udostępniają szeroką gamę usług, dużo przestrzeni dyskowej oraz przepustowości.

Ogólnie rzecz biorąc, firmy zajmujące się udostępnianiem serwerów WWW można podzielić na trzy podstawowe kategorie:

- **Firmy udostępniające serwery dedykowane:** w tym przypadku można wziąć w leasing cały serwer WWW, dzięki czemu tworzona aplikacja będzie mogła w całości wykorzystywać moc procesora, przestrzeń dyskową oraz pamięć serwera. Właściciel ma także pełną kontrolę nad konfiguracją serwera. Rozwiązanie to jest szczególnie korzystne, gdyż właściciel zazwyczaj ma pełną kontrolę nad administracją serwera, a jednocześnie nie musi zaprzętać sobie głowy kupowaniem niezbędnego sprzętu, jego utrzymywaniem, znalezieniem odpowiedniej lokalizacji oraz zapewnieniem połączenia z internetem.
- **Firmy udostępniające serwery współdzielone:** jeśli tworzona witryna WWW nie będzie mieć zbyt dużych wymagań dotyczących zasobów serwera bądź jeśli Czytelnik nie chce zaprzętać sobie głowy administrowaniem serwera, to serwer współdzielony będzie optymalnym rozwiązaniem. Firmy udostępniające serwery współdzielone korzystają z mniejszych wymagań witryn, instalując ich więcej na jednym komputerze i używając wysoce zautomatyzowanych procesów do zarządzania zasobami systemowymi i sieciowymi, tworzenia kopii danych oraz zapewniania wsparcia dla użytkowników. W efekcie ceny takich serwerów mogą być bardzo atrakcyjne (wiele firm cieszących się powszechnym uznaniem udostępnia takie serwery, bez konieczności podpisywania umów, nawet za 8 dol. miesięcznie) przy jednoczesnym zachowaniu pełnej satysfakcji użytkownika.
- **Firmy udostępniające prywatne serwery wirtualne:** prywatne serwery wirtualne zacierają różnice pomiędzy serwerami dedykowanymi i współdzielonymi. Korzystają one z mechanizmu **wirtualizacji**, by udostępniać każdemu użytkownikowi dedykowany system operacyjny, możliwość instalowania na nim aplikacji oraz jego pełnej konfiguracji. Wirtualizacja pozwala na uruchamianie na jednym komputerze wielu niezależnych systemów operacyjnych. W efekcie użytkownik uzyskuje pełną kontrolę nad wykupionym serwerem wirtualnym, natomiast firma udostępniająca serwer może obniżyć koszty, co ma bezpośredni wpływ na ceny świadczonych usług.

Trzeba pamiętać, że poszukiwania i wykupienie serwera na potrzeby tworzonej aplikacji to nie są zadania o szczególnie wysokim priorytecie — nie trzeba zaprzętać sobie tym głowy aż do chwili, gdy aplikacja zostanie ukończona i będzie gotowa do uruchomienia. Dlatego też niezależnie od faktu, że koszty wykupienia serwerów są niskie, warto poczekać z tym aż do momentu, gdy posiadanie serwera będzie absolutnie konieczne, oszczędzając tym samym czas, pieniądze i uwagę.

## Siedem pytań do firm udostępniających serwery WWW

Z pozoru większość firm udostępniających serwery WWW oferuje bardzo podobne usługi. Zapewniają one absurdalnie ogromną ilość przestrzeni na dysku, niemal nieskończoną przepustowość oraz bardzo wysoki gwarantowany czas działania serwera. Szczerze mówiąc, istnieje całkiem spore prawdopodobieństwo, że każda powszechnie szanowana firma udostępniająca serwery WWW w pełni zaspokoi nasze oczekiwania i to zarówno pod względem zasobów wymaganych przez naszą aplikację, jak i pod względem poziomu obsługi technicznej. Niemniej w ramach oceniania usług świadczonych przez poszczególne firmy Czytelnik — jako programista PHP — powinien zadać kilka pytań. Oto one:

1. **Czy jest dostępne środowisko PHP, a jeśli tak, to w jakiej wersji?** Wiele firm udostępniających serwery WWW niezwykle długo ociąga się z instalowaniem najnowszych wersji języka PHP. Dlatego też, jeśli Czytelnik planuje wykorzystać możliwości PHP dostępne od pewnej jego wersji, powinien sprawdzić, czy jest ona dostępna. Dodatkowo optymalnym rozwiązaniem byłoby, gdyby dostawca jednocześnie udostępnił kilka różnych wersji PHP — w takim przypadku można by skorzystać z różnych aplikacji PHP, które dopiero w przyszłości będą obsługiwać najnowsze wersje PHP.
2. **Czy dostępny jest serwer baz danych MySQL/Oracle/PostgreSQL, a jeśli jest, to w jakiej wersji?** Podobnie jak w przypadku PHP, także zainstalowane serwery baz danych nie są zazwyczaj szybko aktualizowane. Dlatego jeśli tworzona aplikacja korzysta z możliwości dostępnych w konkretnej wersji serwera bazy danych, koniecznie należy się upewnić, że dostępna jest odpowiednia wersja.

3. **Jakie są dostępne rozszerzenia plików PHP?** Z niewytłumaczalnych powodów niektóre firmy udostępniające serwery WWW zmuszają użytkowników do stosowania starych rozszerzeń plików PHP — takich jak *.html*. Jest to wyraźnym sygnałem, że danej firmie brakuje rozeznania w zakresie rozwoju języka PHP oraz społeczności jego użytkowników. Oznacza to także, że z usług takiej firmy raczej należy zrezygnować. Warto interesować się jedynie firmami, które pozwalają na stosowanie w skryptach PHP rozszerzenia *.php*.
4. **Jakie ograniczenia są nakładane na skrypty PHP?** Zgodnie z informacjami podanymi we wcześniejszej części rozdziału sposób działania środowiska PHP oraz jego możliwości można określać za pomocą pliku konfiguracyjnego *php.ini*. Niektóre z tych możliwości konfiguracyjnych zostały stworzone dla wygody dostawców serwerów WWW, a ci nie zawsze muszą być chętni, by oddawać w ręce użytkowników wszystkie możliwości, jakimi dysponuje PHP. Dlatego też niektóre funkcje języka oraz jego rozszerzenia mogą nie być dostępne, co w efekcie może mieć wpływ na możliwości, jakie witryna będzie w stanie udostępnić.  
Co więcej, czasami zdarza się, że wszystkie skrypty PHP muszą być umieszczane w konkretnym katalogu — takie rozwiązanie może być bardzo niewygodne, a jego wpływ na zwiększenie bezpieczeństwa aplikacji jest raczej wątpliwy. W optymalnym przypadku powinniśmy mieć możliwość umieszczania skryptów PHP w dowolnym miejscu struktury plików, przeznaczonej do przechowywania stron WWW.
5. **Jakim ograniczeniom podlegają możliwości stosowania plików *.htaccess* serwera Apache?** Niektóre oprogramowania firm niezależnych (w szczególności dotyczy to szkieletów do tworzenia aplikacji, opisanych w rozdziale 24.) do poprawnego działania wymagają tak zwanego **przepisywania adresów URL** (ang. *URL rewriting*). Jednak nie wszystkie firmy udostępniające serwery WWW pozwalają użytkownikom modyfikować działanie serwera Apache przy użyciu specjalnych plików konfiguracyjnych *.htaccess*. Dlatego warto dowiedzieć się, czy dana firma narzuca jakieś ograniczenia w tym zakresie.
6. **Jakie aplikacje PHP są dostępne oraz na jakie wsparcie ze strony firm można liczyć?** Większość firm udostępniających serwery WWW pozwala na automatyczną instalację oprogramowania firm niezależnych, takich jak Joomla!, WordPress czy też phpBB. Możliwość skorzystania z takich instalatorów pozwoli nam zaoszczędzić czas, a firmie obsługującej serwer — rozwiązać ewentualne problemy. Jednak zdarza się, że firmy udostępniają takie oprogramowanie w ramach wygody użytkownika, lecz nie zapewniają żadnej pomocy technicznej związanej z ich instalacją i korzystaniem. Dodatkowo warto zapytać, czy w razie potrzeby firma będzie skłonna zainstalować rozszerzenia PEAR i PECL (zostały one opisane w rozdziale 11.).
7. **Czy na państwa serwerze prawidłowo działa (tu należy wstawić nazwę ulubionego szkieletu aplikacji lub technologii)?** Jeśli Czytelnik planuje korzystać z konkretnego szkieletu do tworzenia aplikacji PHP (więcej informacji na ich temat zostało zamieszczonych w rozdziale 24.) bądź z pewnej technologii (takiej jak określona platforma do handlu elektronicznego), to należy się upewnić, że będą one działały prawidłowo na danym serwerze. Jeśli firma nie jest w stanie udzielić konkretnej odpowiedzi, to można poszukać informacji na forach dyskusyjnych, używając nazwy firmy i technologii jako kryteriów wyszukiwania.

## Podsumowanie

W tym rozdziale został opisany sposób konfiguracji środowiska pozwalającego na tworzenie aplikacji internetowych w języku PHP. Szczegółowo zostały także opisane wybrane opcje konfiguracyjne PHP. W końcowej części rozdziału zamieszczono krótkie informacje dotyczące edytorów i środowisk programistycznych najczęściej używanych do tworzenia aplikacji PHP oraz przedstawiono kilka zagadnień, o których należy pamiętać, poszukując serwera, na którym ta aplikacja będzie działać.

W następnym rozdziale Czytelnik zacznie poznawać język PHP — stworzy swoją pierwszą stronę WWW wykorzystującą PHP oraz pozna podstawowe cechy tego języka. Pod koniec rozdziału Czytelnik będzie w stanie tworzyć proste, lecz całkiem użyteczne skrypty. Zamieszczone w nim informacje będą stanowić podstawę do zagadnień opisanych w kolejnych rozdziałach, które są niezbędne do tworzenia naprawdę fajnych aplikacji.

# Skorowidz

4D, 558

## A

ACID, 646, 649  
Active Directory, 32  
ActivePerl, 243  
ActiveState, 243  
Adobe Dreamweaver CS5, 59  
Adobe Flash, 32  
Ajax, 59, 403, 404, 405, 409, 411, 412  
    Asynchronous JavaScript i XML, 403  
aktualizacja  
    oprogramowanie serwera, 512  
alias  
    zastępcza nazwa domeny, 310  
Amazon, 379, 381  
    FWS, 379  
Apache, 29, 33, 286, 302  
    serwer, 30  
API, 152, 178, 379, 380, 388  
    interfejs programistyczny, 537  
    interfejs programowania aplikacji, 152  
aplikacja  
    i18n, 415  
    internetowa, 403, 405, 537, 591, 598, 601, 610, 611  
    kliencka, 538  
    lokalizacja, 415  
    phpMyAdmin, 603  
    szkielet, 425  
    transakcyjna, 650  
    umiędzynarodowienie, 415  
apostrof, 71, 92, 206, 207, 275, 276, 279  
    lewy, 242, 244  
apt-get, 36  
architektura n-warstwowa, 574  
    n-tier architecture, 574  
ARCHIVE, 442

archiwum  
    tar, 453, 454  
    zip, 453  
ASP, 47  
    składnia, 65  
atak, 273, 512  
    cross-site scripting, 274, 277  
    SQL injection, 537  
    usuwanie plików, 273  
    zapisanie informacji z ciasteczka, 275

## B

baza danych, 32, 126, 156, 274, 291, 292, 423, 425, 435, 436  
    Adabas D, 32  
    administratorzy, 592  
    Berkeley DB, 32  
    database abstraction layer, 555  
    dBase, 32  
    Express, 32  
    FilePro, 32  
    FontBase, 32  
    Hyperwave, 32  
    IBM DB2, 32  
    indeksacja, 631  
    INFORMATION\_SCHEMA, 597  
    Informix, 32  
    Ingres, 32  
    InterBase, 32  
    konto, 393  
    Microsoft SQL Server, 32  
    mSQL, 32  
    MySQL, 32, 60, 61, 286, 291, 292, 293, 294, 441, 537, 613,  
        617, 629, 650  
    Oracle, 32, 60, 61, 555, 653  
    Ovrimos, 32  
    pielęgnacja, 625  
    początkowa, 453  
    PostgreSQL, 32, 60, 61

baza danych  
 relacyjna, 339, 441, 447, 449, 485  
 sekwencja operacji, 645  
 Solid, 32  
 SQLite, 60, 341  
 sterownik, 555  
 struktura, 650  
 Sybase, 32  
 system wieloprocesorowy, 442  
 system wielowątkowy, 442  
 test, 468  
 Unix dbm, 32  
 Velocis, 32  
 warstwy abstrakcji, 555  
 wymuszanie integralności danych, 591  
 wymuszanie reguł biznesowych, 591  
 zajście zdarzenia, 591  
 zapobieganie wprowadzaniu nieprawidłowych danych, 591  
 zapytania, 601

bezpieczeństwo  
 luki procesu, 573  
 opcje, 606  
 polecenie LOAD DATA INFILE, 657

Bind, 33

Bison, 38

BLACKHOLE, 442

blok  
 BEGIN – END, 580  
 case, 96

Blowfish, 29

błąd  
 ignorowanie, 178  
 komunikaty, 302, 304, 561, 442  
 krytyczny, 146, 151, 152, 153, 157, 177  
 obsługa, 30, 560  
 pobieranie kodów SQL, 561  
 połączenia, 560  
 poprawianie, 424  
 poziom zgłaszania, 176  
 rejestracja, 175  
 uruchamiania, 177  
 wyświetlanie w przeglądarce, 177  
 zapisywanie w zmiennej, 178  
 zgłaszanie, 30, 181

BrightHouse, 442

buforowanie  
 danych wyjściowych, 48  
 output buffering, 48  
 wielkości, 48, 49

Ubuntu, 36

## C

C, 31, 63, 67, 68, 77  
 język, 28  
 C#, 30, 149, 175, 181, 279, 280

C++, 30, 63, 66, 67, 147, 181, 442

CGI  
 skrypt, 28  
 CGI 1.1, 82

ciasteczko, 54, 56, 84, 272, 274, 275, 339, 340, 342  
 cookie, 48, 53, 54, 83

CLI, 309  
 Command Line Interface, 309

CloudFront, 379

COM/DCOM  
 obsługa, 29

Content-Type, 57

CPAN, 245  
 Comprehensive Perl Archive Network, 245

CSS, 59, 281, 404, 406, 407, 408, 410, 614, 615, 616  
 kaskadowe arkusze stylów, 614

CSV, 233, 234

czuźysłów, 71, 91, 93, 206, 233, 242, 275  
 automatyczne poprzedzanie, 55  
 magiczny, 55, 56  
 włączanie, 55

CVS, 442

czas, 255, 256, 259, 263, 265, 267, 268  
 Coordinated Universal Time, 255  
 Daylight Saving Time, 255  
 epoka Uniksa, 255  
 formatowanie, 256, 261, 262, 263  
 internetowy, 256  
 letni, 255, 259  
 lokalizacja, 263  
 modyfikowanie, 268  
 na serwerze, 256  
 określanie aktualnego znacznika, 260  
 operacje związane, 258  
 ostatniej modyfikacji pliku, 228, 229  
 pobieranie dokładniejszych informacji, 259  
 Swatch Beats, 257  
 Swatch Internet Time, 256, 257  
 tworzenie znacznika czasu, 261  
 uderzenia Swatch, 257  
 uniwersalny koordynowany, 255  
 Unix epoch, 255  
 utworzenie obiektu, 268  
 wyświetlanie zlokalizowanych dat i czasu, 261  
 częstotliwość występowania wartości, 127

## D

dane  
 dynamiczne, 69  
 eksportowanie, 653, 661, 662  
 importowanie, 653, 654  
 kompresja, 49  
 output buffering, 288  
 początkowe, 454, 455

- separacja poszczególnych elementów, 654
- sesyjne, 339, 340, 341, 344, 346, 347, 351, 352, 355
- skompresowane, 442
- sposób prezentacji, 420
- sposób zarządzania, 442
- statyczne, 69
- uwierzytelniające, 285, 286, 288, 289, 292
- wczytywanie przy użyciu skryptu PHP, 660
- wejściowe, 241
  - kontrola, 223, 241
  - konwersja, 271, 275, 276
  - ograniczenie, 242
  - weryfikacja, 277, 393
  - zabezpieczenie, 241, 393
- wyjściowe, 48, 49, 67, 288
- data, 255, 256, 257, 258, 263, 267, 268
  - formatowanie, 256, 262, 263, 267
  - liczba dni w bieżącym miesiącu, 265
  - liczba dni w podanym miesiącu, 265
  - lokalizacja, 263
  - modyfikowanie, 268
  - obliczanie różnicy, 268
  - ostatniej modyfikacji witryny, 265
  - parametry, 257, 263, 264
  - utworzenie obiektu, 267
  - weryfikacja, 256
  - wyliczenie na podstawie różnicy w dniach, 266
- demon, 178
  - serwera, 45
- deserializacja, 49
- deskryptor
  - private, 151
  - zasięg właściwości, 176
  - zasięgu, 144, 150
- destruktor, 30, 153, 156
  - \_\_destruct(), 156
- DN
  - rozpoznawalna nazwa, 330, 331, 335, 336
- DNS, 32, 309, 310, 311, 312
  - Domain Name System, 310
  - nazwy typów, 312
  - pobieranie rekordów, 311, 313
  - rekord, 310, 311
  - sprawdzenie istnienia rekordu, 310
  - typy rekordów, 310, 312
- DocBlocks, 66
- DOM, 403, 404, 405, 407, 409
  - model obiektów dokumentu, 404
- domknięcia
  - closure, 30
- DSN, 293
  - Data Source Name 293
- dyrektywa
  - AddType, 39, 40
  - allow\_call\_time\_pass\_reference, 49
  - allow\_url\_fopen, 58, 102
  - always\_populate\_raw\_post\_data, 57
  - arg\_separator.input, 54
  - arg\_separator.output, 53
  - asp\_tags, 47
  - auto\_append\_file, 56
  - auto\_detect\_line\_endings, 59
  - auto\_prepend\_file, 56
  - crack.default\_dictionary, 295, 296
  - date.timezone, 256
  - default\_charset, 57
  - default\_mimetype, 56
  - default\_socket\_timeout, 59
  - DirectoryIndex, 43
  - disable\_classes, 51, 394
  - disable\_functions, 51, 394
  - display\_errors, 177, 394
  - display\_startup\_errors, 177
  - doc\_root, 57
  - DocumentRoot, 51
  - engine, 47
  - error\_log, 178
  - error\_reporting, 176, 177
  - expose\_php, 52, 53
  - extension, 44, 58, 59
  - extension\_dir, 44, 58
  - file\_uploads, 301
  - fopen\_wrappers, 236
  - from, 58
  - highlight.bg, 52
  - highlight.comment, 52
  - highlight.default, 52
  - highlight.html, 52
  - highlight.keyword, 52
  - highlight.string, 52
  - ignore\_repeated\_errors, 178
  - ignore\_repeated\_source, 178
  - ignore\_user\_abort, 51, 52
  - implicit\_flush, 49
  - include\_path, 57, 231, 232, 236, 248, 251, 252
  - konfiguracja, 44, 45, 46, 49, 51, 52, 53, 58, 175, 176, 316, 340
  - LoadModule, 40
  - log\_errors, 178
  - log\_errors\_max\_len, 178
  - magic\_quotes\_gpc, 55, 56
  - magic\_quotes\_runtime, 56, 237
  - max\_execution\_time, 46, 53, 395
  - max\_file\_uploads, 301
  - max\_input\_time, 53, 301
  - memory\_limit, 53, 301, 395
  - obsługa sesji, 340
  - open\_basedir, 51, 395
  - output\_buffering, 48
  - output\_handler, 48, 49
  - PHP\_INI\_ALL, 47, 48, 49, 51, 52, 53, 54, 56, 57, 58, 59
  - PHPIniDir, 44



## dyrektywa

- post\_max\_size, 56, 301
  - precision, 47, 49
  - przesyłanie plików, 300
  - register\_long\_arrays, 55
  - safe\_mode, 50, 51
  - safe\_mode\_allowed\_env\_vars, 50
  - safe\_mode\_protected\_env\_vars, 51
  - safe\_mode\_exec\_dir, 50
  - safe\_mode\_gid, 50
  - safe\_mode\_include\_dir, 50
  - ServerSignature, 53
  - short\_open\_tag, 45, 46, 47, 64
  - SMTP, 317
  - sql.safe\_mode, 395
  - track\_errors, 178
  - track\_vars, 82
  - unserialize\_callback\_func, 49
  - upload\_max\_filesize, 301
  - upload\_tmp\_dir, 301, 302
  - user\_agent, 58
  - user\_dir, 58
  - variables\_order, 54, 84
  - wartości, 45
  - y2k\_compliance, 48
  - zasięg, 46
  - zend.ze1\_compatibility\_mode, 47
  - zlib.output\_compression, 48, 49
- dyrektywa konfiguracyjna, 394
- bezpieczeństwo, 394
- dysk
- określanie całkowitej pojemności, 226
  - określanie wielkości, 225, 226, 227
  - określanie wolnego miejsca, 226
- dziedziczenie, 141, 142, 143, 153, 160, 161, 164, 165, 166, 171, 292, 293
- wiązanie statyczne, 168
  - wielokrotne, 162
- dziennik
- systemowy, 180
  - zdarzeń, 178

## E

- Eclipse, 60
- edytor
  - programistyczny, 35, 59
  - tekstów, 41
- eksport
  - danych, 653
- E-Soft, 30
- EXAMPLE, 442

## F

- Falcon, 442
- FastCGI, 29, 41
- FEDERATED, 442
- Firebird / Interbase, 558
- Flex, 38
- format
  - bzip2, 37
  - CHM, 37
  - DocBook, 37
  - gzip, 37
  - HTML, 37
  - PDF, 37
  - XML, 37
- formularz
  - grupowanie pól, 282
  - HTML, 393
  - tworzenie i weryfikacja danych, 281
  - weryfikacja danych, 273
  - złożone komponenty, 278
- forum, 423
- FTP, 58, 230, 299, 308, 313
- funkcja, 66, 78, 105, 106, 107, 108, 109, 110, 113, 176
  - \_autoload, 159
  - addslashes(), 55
  - amortizationTable(), 110
  - argumenty opcjonalne, 108
  - array\_chunk(), 140
  - array\_combine(), 135
  - array\_count\_values, 219
  - array\_count\_values(), 127
  - array\_diff, 137
  - array\_diff\_assoc(), 138
  - array\_flip(), 129, 209
  - array\_intersect(), 136
  - array\_intersect\_key(), 137
  - array\_intersect\_ukey(), 137
  - array\_key\_exists(), 122
  - array\_merge(), 134
  - array\_merge\_recursive(), 134
  - array\_push(), 121
  - array\_rand(), 139
  - array\_reverse(), 129
  - array\_search(), 123
  - array\_shift(), 121
  - array\_slice(), 135
  - array\_splice(), 136
  - array\_sum(), 139
  - array\_udiff(), 138
  - array\_unique(), 128
  - array\_unshift(), 121
  - array\_values(), 123, 198
  - array\_walk(), 126
  - array\_walk\_recursive (), 127

arsort(), 131  
 asort(), 130, 131  
 assignx(), 78  
 basename(), 224  
 biblioteka, 105, 113, 158  
 biblioteka do obsługi dat i czasu, 256  
 bindtextdomain(), 416, 417  
 calcSalesTax(), 107, 109  
 calculate(), 109  
 checkdate(), 256  
 checkdnsrr(), 310, 312  
 chgrp(), 50  
 chmod(), 50  
 chown(), 50  
 class\_alias(), 158  
 class\_exists(), 158  
 closedir(), 238  
 closelog(), 180  
 copy(), 303  
 count(), 127  
 count\_chars(), 217  
 crack\_check()., 296  
 crack\_opendict(), 296  
 current(), 124  
 date(), 256, 257, 258, 260, 263, 265, 267  
 date\_default\_timezone\_set(), 256  
 define(), 84  
 define\_syslog\_variables(), 179  
 definicja, 67, 113  
 deklarowanie argumentów, 79  
 destruktory, 153  
 dirname(), 224  
 disk\_free\_space(), 226  
 disk\_total\_space(), 226, 227  
 dns\_get\_record(), 311, 312  
 dokumentacja, 106  
 domyślne wartości argumentów, 108  
 echo(), 49, 63, 125  
 enable\_dl(), 58  
 end(), 125  
 ereg(), 192, 194  
 ereg\_replace(), 193, 194  
 error\_get\_last(), 177  
 escapeshellarg(), 241, 242, 275  
 escapeshellcmd(), 241, 275, 276  
 exec(), 50, 51, 53, 227, 239, 242, 243, 273, 274, 276  
 explode(), 118, 211, 212, 290  
 fclose(), 231  
 feof()., 229  
 fgetc(), 235  
 fgetcsv(), 233, 660  
 fgets(), 59, 234  
 file(), 50, 52, 59, 229, 231, 232, 233, 234, 290  
 file\_get\_contents()., 232  
 fileatime(), 228  
 filectime(), 228  
 filemtime(), 229  
 filesize(), 226, 235  
 filter\_var(), 277, 278  
 fopen(), 230, 231, 232, 238, 626  
 fopen\_with\_path()., 57  
 fread(), 235  
 fscanf(), 236  
 fseek(), 237  
 fsockopen(), 231, 314, 315  
 fwrite(), 236  
 generateFooter(), 106  
 get\_class(), 158  
 get\_class\_methods(), 158  
 get\_class\_vars(), 158  
 get\_declared\_classes(), 158  
 get\_html\_translation\_table(), 208  
 get\_object\_vars(), 159  
 get\_parent\_class(), 159  
 getdate(), 259  
 getmxrr(), 313  
 getservbyname(), 313  
 getservbyport(), 313, 314  
 gettext(), 416  
 gettimeofday(), 259  
 gettype(), 75  
 header(), 288  
 highlight\_file(), 52  
 htmlentities(), 206, 207, 275, 276, 277  
 htmlspecialchars(), 208  
 htmlstriptags(), 208  
 implode(), 212  
 in\_array(), 122  
 include(), 56, 57  
 informacja o typie, 109  
 ini\_set(), 46  
 interface\_exists(), 159  
 is\_a(), 159  
 is\_array(), 75, 119  
 is\_bool(), 75  
 is\_float(), 75  
 is\_integer, 75  
 is\_null(), 75  
 is\_numeric(), 75  
 is\_object, 75  
 is\_resource, 75  
 is\_scalar(), 75  
 is\_string, 75  
 is\_subclass\_of(), 159  
 is\_uploaded\_file(), 299, 302, 303  
 isset(), 288  
 key(), 124  
 konstruktory, 153  
 krsort(), 132  
 ksort(), 132

## funkcja

ldap\_add(), 332  
 ldap\_bind(), 327, 328  
 ldap\_close(), 328  
 ldap\_search(), 329  
 ldap\_start\_tls(), 327  
 ldap\_unbind(), 328  
 ltrim(), 214, 216  
 mail(), 316, 317  
 matematyczna, 87  
 mb\_convert\_case(), 205  
 md5(), 290, 297  
 method\_exists(), 159  
 mktime(), 260, 261, 266  
 move\_uploaded\_file(), 299, 302, 303  
 mysql\_connect(), 395  
 mysql\_pconnect(), 395  
 natsort(), 129, 131, 132  
 next(), 124, 125  
 nl2br(), 206  
 ob\_gzhandler(), 48  
 obsługa przesyłanych plików, 302  
 określanie typu zmiennych, 75  
 opendir(), 238  
 openlog(), 179, 180  
 parametry, 78, 79, 107, 109  
 parse\_ini\_file(), 50  
 passthru(), 241, 243  
 pathinfo(), 224, 225  
 phpinfo(), 42  
 pomocnicza, 158  
 pop, 121  
 pow(), 105, 106  
 preg\_filter(), 200  
 preg\_grep(), 197  
 preg\_match(), 198  
 preg\_match\_all(), 198  
 preg\_quote(), 199  
 preg\_replace(), 199  
 preg\_split(), 201  
 prev(), 125  
 print(), 49, 63, 68  
 print\_r(), 120  
 printf(), 63, 68, 75  
 processPayPalPayment(), 109  
 przekazywanie argumentu przez referencję, 107  
 przekazywanie argumentu przez wartość, 107  
 przekazywanie danych, 107  
 przekazywanie wartości z tablicy, 126  
 push, 121  
 range(), 118, 119  
 readdir(), 238  
 readfile(), 236  
 realpath(), 225  
 rekurencyjna, 80, 105, 110  
 RemoveProfanity(), 172  
 rename(), 240  
 require(), 56, 57, 105  
 rmdir(), 239  
 rsort(), 130  
 rtrim(), 216  
 scandir(), 239  
 setcookie(), 83  
 setlocale(), 128, 262, 263, 264, 416, 419, 420  
 settype(), 75  
 shell\_exec(), 244  
 shift, 121  
 show\_source(), 52  
 shuffle(), 139  
 sizeof(), 127, 211  
 sort(), 129, 130, 131, 132  
 split(), 194, 290  
 spliti(), 195  
 sprintf(), 63  
 sql\_regcase(), 195  
 str\_pad(), 217  
 str\_replace(), 194, 213  
 str\_word\_count(), 218  
 strcasecmp(), 203  
 strcmp(), 202, 203  
 strcspn(), 204  
 strftime(), 262, 263, 264  
 strip\_tags(), 126, 209, 211, 235, 275, 277  
 stripos(), 212  
 stripslashes(), 55  
 strpos(), 212  
 strrpos(), 213  
 strspn(), 203  
 strstr(), 214  
 strtok(), 210  
 strtolower(), 204  
 strtotime(), 266, 267  
 strtoupper(), 204, 205  
 strtr(), 209  
 substr(), 214  
 substr\_count(), 215  
 substr\_replace(), 216  
 syslog(), 180  
 system(), 50, 51, 53, 227, 239, 243, 273  
 textdomain(), 416  
 time(), 260  
 touch(), 240  
 trim(), 217  
 tworzenie, 106  
 ucfirst(), 205  
 ucwords(), 205  
 uniqid(), 297  
 unshift, 121  
 usort(), 132  
 vprint(), 119

vsprintf(), 120  
 wejścia-wyjścia, 236  
 wykonywanie programów, 242  
 wynik wywołania, 106  
 wywołanie, 105, 106, 107, 108  
 zastępująca, 200  
 zmiennej, 105  
 zwracanie wartości z funkcji, 109  
 zwracanie wielu wartości, 110

## G

Garrett Jesse James, 403  
 Gettext, 416, 418, 419, 420  
 GIF, 357  
 global, 79  
 globalna przestrzeń nazw, 54  
 gniazdo, 59  
 GNU, 359, 453, 454, 455  
   Lesser General Public License, 359  
   GPL, 60  
 GPL, 243  
 grupa interfejsów, 146  
 GUI, 467  
   graficzny interfejs użytkownika, 467  
 Gutmans Andi, 28

## H

hasło, 285, 393  
 biblioteka CrackLib, 294  
 długość, 295  
 jednorazowe adresy URL, 285, 296, 297, 298  
 łamanie metodą brutalnej siły, 294  
 odzyskiwanie, 296  
 podobieństwo, 295  
 proces akceptacji, 294  
 rozszerzenia CrackLib, 295  
 standardowa numeracja, 295  
 testowania siły, 294  
 wielkości liter, 295  
 wyczyszczenie, 298  
 zróżnicowanie, 295  
 hermetyzacja, 141, 143, 146  
 danych wewnątrz obiektu, 146  
 encapsulation, 142  
 zapytań, 601  
 HTML,  
   formularz, 28, 32, 299, 300, 304  
   konwersja, 202, 206, 208, 209  
   obsługa formularzy, 271  
   przesyłanie danych, 300  
   tabela, 614  
   tabela na podstawie informacji z bazy danych, 617  
   wpisywanie kodu, 208

wysyłanie wiadomości, 319  
 znacznik, 126, 190, 199, 206, 207, 208, 209, 234, 235  
 HTML\_Common, 615  
 HTML\_Common2, 281  
 HTML\_QuickForm2, 281  
   konstruktor klasy, 282  
 HTML\_Table, 614, 615, 616  
   tworzenie prostej tabeli, 615  
 HTMLDOC, 241, 242  
 HTTP, 56, 230, 299, 303, 305, 306, 308, 313  
   Hypertext Transfer Protocol, 339  
   nagłówek, 288  
   protokół, 339, 380, 381  
   przesył hipertekstu, 339  
 http\_Upload, 306, 307  
 HTTPS, 230, 380  
 Hyde Randall, 91

## I

IBM DB2, 558  
 IDE, 35, 60  
 identyfikator, 76  
   filtru, 278  
   generowanie unikalnego, 297  
   GID, 50  
   grupy, 50  
   nazwa, 76  
   połączenia, 328  
   sesji, 54, 339, 340, 342, 343, 344, 345, 348  
   session ID, 339  
   session identifier, 339  
   SID, 339  
   UID, 50  
   użytkownika, 50, 297  
 IIS  
   Microsoft Internet Information Server, 286  
 IIS Developer Express, 41  
 IMAP, 32, 292, 313  
 implementacja  
   kilku interfejsów, 170  
   pojedynczego interfejsu, 170  
 indeks, 198  
   asocjacyjny, 72  
   główny, 632, 633, 643  
   liczbowy, 72  
   normalny, 634  
   normalny wielokolumnowy, 634  
   pełnotekstowy, 635, 636  
   unikalny, 633  
   założony wielokolumnowy, 635  
 indeks numeryczny, 121  
   wartość, 116  
 indeksacja, 631, 639, 643  
   leftmost prefixing, 634  
 indeksowanie liczbami, 121, 134

- Infobright, 442
  - informacja o typie, 105
    - type hinting, 105
  - Informix, 558
  - inkrementacja, 96
  - InnoDB, 442, 446
    - mechanizm składowania danych, 646, 647, 649, 650
  - instalacja
    - custom (dowolna), 39
    - typical (typowa), 39
  - instalator dla systemu Windows
    - Windows installer, 37
  - instancja
    - niezdefiniowanej klasy, 49
  - instrukcja
    - \$count, 97
    - break, 96, 99
    - case, 158
    - catch, 181
    - continue, 100
    - do...while, 97, 99
    - dołączanie plików, 101
    - echo(), 63, 68, 92, 101
    - else, 94
    - elseif, 95
    - for, 94, 97, 99
    - foreach, 94, 98, 99
    - goto, 99, 100
    - if, 89, 94, 101, 128, 358
    - include(), 101, 102, 105, 113
    - include\_once, 102
    - print(), 67, 68, 92
    - printf(), 69, 70
    - require, 102, 287
    - require\_once, 103
    - require\_once(), 160
    - return(), 109
    - sekwencja, 96
    - sprintf (), 70
    - switch, 94, 95, 96, 99, 158
    - try, 181
    - warunkowa, 89, 94, 581
    - while, 94, 96, 99
  - interfejs, 161, 168, 169, 170, 171
    - aplikacji, 358, 403, 404
    - ilość, 162
    - obiektowy, 556
    - obiektowy dla dat i czasu, 266
    - programistyczny, 467
    - sprawdzenie, 159
    - SQL, 555
  - inventory\_manager, 273, 274, 276
  - ISAPI
    - obsługa, 29
  - ISO, 416
    - Międzynarodowa Organizacja Normalizacyjna, 416
  - ISO-8859-1, 207, 419
- ## J
- Java, 30, 147, 149, 175, 181, 300, 442
    - język, 29
    - obiekty, 29
  - JavaScript, 59, 274, 277, 357, 371, 403, 404, 405, 407, 411, 412
    - asynchroniczne, 403
    - biblioteka, 404
  - Joomla!, 62
  - jQuery, 403, 404, 405, 406, 407, 408, 409, 411, 412
    - biblioteka, 405
    - prezentacja, 405
  - JSON, 404, 411, 412
    - format, 404
    - JavaScript Object Notation, 404
- ## K
- kanal
    - feed, 381
    - URL, 381
  - katalog
    - analizowanie zawartości, 238
    - chroniony, 287
    - docelowy, 305
    - docs, 37
    - htdocs, 41
    - instalacyjny, 37, 38
    - języka, 419
    - lista plików, 243
    - manual, 37
    - niewłaściwe uprawnienia, 305
    - obsługa zawartości, 223
    - określanie ścieżki, 224
    - określanie wielkości, 224, 225, 227, 238, 239
    - określanie zawartości, 238
    - operacje na plikach, 395, 398
    - PEAR, 251
    - pobieranie informacji, 224, 231, 236
    - prawo do zapisu, 239
    - przeszukiwanie, 329
    - tyczasowe pliki, 302
    - tyczasowy, 302, 305
    - uchwyt, 238
    - usuwanie, 239
    - wczytanie zawartości, 239
    - zawartość, 242
  - kiosk, 538
  - klasa, 66, 142, 144, 145, 146, 151, 152, 153, 154, 155, 157, 159, 160
    - abstrakcyjna, 30, 152, 161, 169, 171, 172
    - alias, 158

- automatyczne wczytywanie, 159
- bazowa, 152, 153, 155, 159, 161, 165, 167, 171, 172, 182, 184
- biblioteka, 172, 394
- cecha, 142
- chroniona, 151
- definicja, 73, 142, 143, 163
- deklarowanie właściwości, 144
- dziedziczenie, 165
- graph, 51
- hierarchia, 161, 168
- implementacja interfejsu, 169
- instancje, 144
- konkretna, 161
- metody, 150
- nadrzędna, 159, 165
- nazwa, 154, 157, 158
- niezdefiniowana, 160
- obiekt, 149, 151, 155, 156, 157, 159, 173
- odwołanie się do właściwości, 145, 147
- pochodna, 146, 151, 152, 155, 157, 159, 165, 168
- podrzędna, 165
- przeciążanie właściwości, 147
- rozszerzenie definicji, 148
- składowe statyczne, 156
- stałe, 149
- tworzenie, 143
- tworzenie instancji, 144
- tworzenie nazwy zastępczej, 158
- vector, 51
- właściwości, 144
- właściwości sfinalizowane, 147
- wywołanie konstruktora, 155
- zasięg prywatny, 145, 146
- zasięg chroniony, 145
- zasięg finalny, 145
- zasięg statyczny, 145
- zasięg właściwości, 145
- zastępcza, 158
- zmiana nazw, 153
- klient
  - mysql, 467, 470, 471, 474, 475, 476
  - mysqlimport, 658, 659
- klucz
  - asocjacyjny, 116, 121, 122, 134
  - numeryczny, 116
  - PRIMARY, 631
  - UNIQUE, 631
  - wartości, 632
- kod
  - bloki, 64, 65
  - błędu, 83
  - HTML, 63, 64
  - jednowierszowe komentarze, 66
  - kolorowanie, 52
  - komentarze wielowierszowe, 67
  - komentowanie, 63, 66
  - PHP, 64
  - umieszczanie, 63, 65
  - źródłowy PHP, 38
  - źródłowe serwera Apache, 38
- kolejka
  - definicja, 121
  - FIFO, 121
  - first in, first out, 121
  - LIFO, 121
- kolumna
  - baza danych, 485
  - definicja, 633
- kompilator
  - ANSI-C, 38
- komunikacja
  - asynchroniczna, 405
- konfiguracja, 394
- konkatenacja, 88
- konstrukcja
  - array(), 117
  - list(), 110, 118, 234
- konstruktory, 30, 153, 166
  - \_construct, 154
  - klasa bazowa, 166, 168
  - klasa DateTime, 267
  - klasa pochodna, 167
  - nazwa, 154
  - niewiązany, 155
  - wywołanie, 167
- konto
  - root, 38
  - superużytkownika, 38
- kontroler, 425, 433
  - fasady, 425
  - front controller, 425
- konwersja
  - formatów liczbowych, 246
  - liczb arabskich na rzymskie, 245
  - między formatami, 654
  - znacznika czasu, 259
  - znaków nowego wiersza na znaczniki br, 206
  - znaków specjalnych, 206
- konwerter
  - podsieci, 322
- kursor
  - obsługa po stronie serwera, 626
  - odczyt tylko w jednym kierunku — do przodu, 627
  - otwieranie, 627
  - przeznaczenie tylko do odczytu, 626
  - stosowanie, 628
  - wrażliwość, 626
  - zamykanie, 628
- kwantyfikator, 191

## L

lambda  
funkcje, 30

LDAP, 32, 292, 324, 325, 326, 327, 328, 329, 330, 331, 333, 334, 335, 336, 337  
IBM Redbooks, 326  
implementacja, 326  
Lightweight Directory Access Protocol, 326  
operacje na zwróconych rekordach, 330  
pobieranie konkretnego rekordu, 331  
port, 327  
poszukiwanie rekordu, 329  
rekordy, 332  
serwer, 326  
sortowanie rekordów, 332  
specyfikacja, 326  
wpis do katalogu, 332  
wstawianie danych, 332  
zamykanie połączenia z serwerem, 328

LDAPS  
Secure LDAP, 327

Lerdorf Rasmus, 28, 31

LGPL, 359

libxml2  
biblioteka, 30

licznik ruchu, 27

localhost, 39, 315, 316

lokalizacja  
l10n, 415

loosely typed, 32

## Ł

łańcuch  
CHARSET, 418  
wielobajtowy, 205

łańcuch znaków  
analiza, 212  
dopełnianie, 202, 216, 217  
dzielenie, 211  
inne operacje, 201  
konwersja, 202, 206  
odczytywanie zawartości pliku, 232, 236  
odnajdywanie ostatniego wystąpienia, 213  
określanie długości, 202  
określanie liczby znaków, 217  
pobieranie fragmentu, 192, 203, 204, 214, 215  
podobieństwo, 203  
podział, 194, 195, 201, 210  
na fragmenty, 201  
porównanie, 202  
przeszukiwanie, 197  
przycinanie, 202, 216, 217  
usuwanie znaków, 216, 217

wyliczanie różnicy, 199, 204  
zapisywanie, 244  
w pliku, 236

zastępowanie  
fragmentu, 216  
wystąpienia, 213

zliczanie  
słów, 218  
znaków i słów, 202, 217

zmiana  
pierwszej litery, 205  
wielkości liter, 202, 204

## M

Mac OS X, 36, 59, 60, 451

Macintosh, 59

MagpieRSS., 379, 382

Masinter Larry, 299

MCrypt  
biblioteka, 29

MD5, 29, 290, 292, 294  
Message-Digest, 290

MDB2, 32

mechanizm  
obsługi szablonów, 358, 357, 360  
templating engines, 357  
rejestracji, 179  
składowania, 442, 447, 485, 486, 489, 491, 492, 493, 494, 507, 509, 646  
storage engines, 442

MEMORY, 442

MERGE, 442

metaznak  
przykład, 196

metoda  
\_clone(), 163  
\_get(), 147, 148, 149  
\_set(), 147, 148, 149  
abstrakcyjna, 152  
akcesor, 148  
calculateSalary(), 150  
DateTime(), 267  
deklarowanie, 150  
errorMsg(), 306  
GET, 82, 84, 102, 272  
getFiles(), 306, 307  
informacje o typach, 153  
isValid(), 306, 308  
moveTo(), 306, 308  
mutator, 147  
niestandardowa  
pobierająca, 149  
ustawiająca, 149  
pobierająca, 148, 162

- pobieranie nazw, 158
  - POST, 82, 83, 84, 272, 282, 301, 303
  - przeciążanie, 162
  - publiczna, 146, 168
  - setCancelText(), 293
  - setName(), 146
  - setPower(), 73
  - sfinalizowana, 152
  - ustawiająca, 147, 162
  - validateCardNumber(), 151
  - wywoływanie, 150
  - zasięg, 150
  - Microsoft Web Platform Installer, 41
  - Microsoft Access, 661
  - Microsoft Active Directory, 325
  - Microsoft Excel, 663
  - Microsoft Office, 299
  - Microsoft SQL Server, 558
  - MIME, 39, 40, 53, 56
    - typ przesłanego pliku, 302
  - model, 423, 425, 437
    - wzór konieczny, 425
  - modyfikator
    - przykład, 196
  - Mozilla Firefox, 286
  - MPEG, 299
  - MS-DOS, 59
  - MSI Installer Essentials, 452
  - MTA, 316, 317
    - Mail Transfer Agent, 316
  - Muffett Alec, 294
  - MVC
    - Model-View-Controller, 423
    - model-widok-kontroler, 423
    - wzorzec, 423, 424, 425, 439
  - MyISAM, 442, 485, 486, 487, 488, 489, 490, 491, 492, 493, 509
  - MySQL
    - AB, 441
    - administracja, 467
    - atrybut tabeli, 485
    - biblioteka osadzonego serwera, 538
    - Community Edition Server, 451
    - data i czas, 485
    - Enterprise Editio Server., 451
    - Improved, 537
    - instalacja, 451, 452
    - instalacja serwera, 451, 452
    - klient, 453, 455, 457, 458, 467, 468, 470, 471, 472, 474, 475, 476, 603, 645
    - kod źródłowy, 452
    - kod źródłowy serwera, 442
    - komponent serwera i klienta, 454
    - konfiguracja serwera, 451, 511
    - mechanizm składowania, 442
    - narzędzia, 652, 654
    - obsługa osadzonego serwera, 538
    - obsługa wyzwalaczy, 594
    - opcje, 468
    - pakiet RPM, 452
    - platformy systemowe, 442
    - plik binarny, 452, 453, 454, 455
    - pobieranie serwera, 451
    - polecenia, 485, 573
    - RPM Package Manager, 452
    - serwer, 442, 467, 573
    - skompilowane kody źródłowe, 453
    - słowo ignorowane, 637
    - sterownik, 30
    - system uprawnień, 511
    - tabela, 485, 655
    - typ
      - liczbowy, 485, 494
      - tekstowy, 485
      - danych, 485
    - wersje, 442
    - wydajność, 442
    - zapytania, 573
  - mysql, 537, 538, 539, 540, 541, 542, 543, 544, 545, 546, 547, 548, 549, 550, 551, 552, 553
    - debugowanie, 538
    - grupy klas, 537
    - obiektość, 537
    - obsługa transakcji, 538
  - mysqlimport
    - opcje, 658
- ## N
- nawias
    - kwadratowy, 190
    - klamrowy, 92
  - nazwa
    - bazy danych, 468, 480
    - hosta, 468, 472, 474, 475, 476, 481
    - użytkownika, 468
    - zastępcza, 601
  - Nebel Ernesto, 299
  - Netcraft, 28
  - Netpbm, 243
  - Netscape Navigator, 286
  - NitroEDB, 442
  - Nmap, 316, 322
  - NNTP, 32
  - Notepad++, 60
  - Novell eDirectory, 325
  - NOWDOC, 30



## O

obiekt, 142, 144, 171  
   definicja klasy, 72  
   klasa, 72  
   klasa abstrakcyjna, 172  
   klonowanie, 30, 161, 162  
   określanie typu, 159  
   słowo kluczowe new, 73  
   tworzenie, 153, 161, 171  
   usuwanie, 153  
 ODBC, 558  
   sterownik, 556  
 odwzorowanie  
   kluczy, 129, 130  
   kluczy i wartości, 130, 131, 140  
 ogranicznik, 358, 373  
   nawias klamrowy, 358, 361  
 określanie kontekstu obiektu, 158  
 opcja  
   --with-config-file-path, 38  
 open source, 33  
 OpenLDAP, 326, 327  
 Opera, 286  
 operacja na wartości, 31  
 operacja na właściwości, 147  
   accessors, 147  
   akcesoria, 147  
   getters, 147  
   metoda  
     pobierająca, 147  
     ustawiająca, 147  
   mulator, 147  
   setters, 147  
 operand, 85  
 operator, 85  
   ->, 157  
   arytmetyczny, 87  
   bitowy, 90  
   dekrementacja, 88  
   inkrementacja, 88  
   konkatenacja, 88, 106  
   logiczny, 89, 90, 638  
   łańcuchowy, 88  
   łączność, 86  
   odwrotnego apostrofu, 50  
   porównania, 89, 90  
   priorytet, 86  
   przeciążanie, 162  
   przypisania, 86, 87, 88  
   przypisania skrótowy, 87  
   równości, 89  
 Oracle, 441, 558, 661  
 Oracle Beehive, 325

## ORM

mapowanie obiektowo-relacyjne, 33  
 object relational mapping, 33  
 Orte Monte, 359

## P

parametr  
   datadir, 647  
   dopasowania, 194, 198  
   konfiguracyjny, 647  
   ogranicznik, 199  
   przesunięcie, 198, 212, 213, 215  
   URL, 393  
   wejściowy, 277  
 PCRE  
   biblioteka, 29, 32  
   pakiet, 29  
   Perl Compatible Regular Expressions, 29  
 PDF, 32, 66, 241, 299, 302, 304  
 PDO, 32  
   elastyczność, 556  
   instalacja, 557  
   interfejs obiektowy, 556  
   obsługa baz danych, 558  
   opcje połączenia, 559  
   PHP Data Object, 30, 556  
   pobieranie danych, 567  
   pobieranie i ustawianie atrybutów, 562  
   połączenie z serwerem, 558  
   spójność, 556  
   wiązanie parametrów, 565  
   wydajność, 557  
   wykonywanie zapytań, 563  
 PDT, 60  
   PHP Development Tools, 60  
 PEAR, 62, 141, 189  
   aktualizacja, 246, 248  
   aktualizacja pakietów, 252  
   Auth\_HTTP, 292  
   automatyczna instalacja, 251  
   dołączanie pakietów do skryptów, 251  
   Extension and Application Repository, 32  
   HTML\_QuickForm2, 280  
   HTTP\_Upload, 305  
   instalacja, 247, 292  
   instalacja i aktualizacja, 245, 246  
   instalacja pakietów Mail oraz Mail\_Mime, 317  
   instalacja wybranego pakietu, 250, 253  
   lokalna instalacja na serwerze, 248  
   menedżer pakietów, 248, 251, 252, 253, 254  
   odinstalowanie pakietu, 253  
   Package Manager, 245, 246, 247, 248  
   Packet Manager, 246  
   pakiet Mail, 309, 317

- pakiet Mail\_Mime, 309, 317
- pakiety, 249
- pakiety domyślne, 247
- PHP Extension and Application Repository, 245
- plik go-pear.bat, 247
- repozytorium, 246, 281, 305
- ręczna instalacja, 251
- tabelaryczna prezentacja danych, 614
- ustawienia konfiguracyjne, 247
- Validate\_US, 189
- PEAR2, 254
- PECL, 62, 295
  - PHP Extension Community Library, 295
- Perl, 33, 36, 38, 189, 190, 195, 196, 197, 233, 242, 243, 300, 309
  - język, 28
- Personal Home Page, 28
- pętla, 96, 582
  - do...while, 97
  - for, 89, 97, 99, 121
  - foreach, 98, 99, 119
  - while, 89, 96, 97, 128
- Phing, 141
- PHP
  - biblioteka, 32, 48
  - blok kodu, 101
  - cechy, 27
  - cena, 33
  - dokumentacja, 37, 66
  - domyślne ustawienia lokalne, 262
  - Extension and Application Repository, 32
  - formularze HTML, 271
  - funkcje, 32
  - funkcje do wykonywania programów 227, 241
  - hasła, 32
  - historia, 27, 28, 33
  - implementacja interfejsu, 169
  - instalacja, 29, 34, 35, 62
  - interpreter, 39, 40, 63, 92
  - język, 27, 28
  - konfiguracja, 34, 35, 38, 40, 43, 44
  - licencja, 33
  - obiektowość, 29
  - obsługa sesji HTTP, 29
  - ograniczenia, 33, 62
  - parser, 28, 84
  - PEAR, 245
  - pliki źródłowe, 37
  - programista, 27
  - programowanie obiektowe, 30
  - przesyłanie plików, 300
  - rozszerzenia plików, 62
  - skalowalność, 28
  - składnia kodu, 31
  - skrypt, 31
  - słowa kluczowe, 76
  - source, 37
  - szkielet do tworzenia aplikacji, 62
  - zsyfrowanie, 29
  - środowisko, 37, 41, 43, 47, 61
  - wersja, 27, 61
  - wysyłanie poczty elektronicznej, 317, 309
  - zarządzanie zasobami, 53
  - znaczniki, 126
- PHP Extension and Application Repository, 144
- PHP/FI
  - Personal Home Page/Form Interpreter, 28
- phpBB, 62
- phpDocumentor, 66, 141
- plik
  - .htaccess, 44, 45, 46, 62, 286, 287
  - .htpasswd, 285, 287
  - biblioteka funkcji, 113
  - binarny, 419
  - CGI, 45
  - CHM, 37
  - domyślny, 43
  - dziennika, 177, 178, 181, 394
  - flat files, 341
  - handle, 230
  - HTML, 37
  - httpd.conf, 39, 40, 43, 44, 45, 46, 52, 53
  - informacja o przesłanym, 306, 307
  - konfiguracyjny, 44, 160
  - lokalny system, 231
  - nazwa, 232
  - odczytywanie
    - zawartości plików, 231
    - zawartości znak po znaku, 235
    - czasu dostępu i modyfikacji, 228
    - czasu modyfikacji, 240
  - określanie wielkości, 225, 226
  - operacja wejścia-wyjścia, 223, 230
  - operacje na plikach, 229
  - otwieranie i zamykanie, 230
  - php.ini, 38, 40, 44, 45, 46, 50, 62, 64, 82, 256, 295, 557, 559
  - php.ini-development, 44, 46
  - php.ini-dist, 38, 44
  - php.ini-production, 44, 46
  - php.ini-recommended, 38, 44
  - pobieranie informacji, 224
  - prosty, 341
  - przenoszenie przesłanych, 303
  - przesuwanie wskaźnika, 237
  - przesyłanie większej liczby, 307
  - przetwarzanie ścieżki, 224
  - pyrus.phar, 253
  - rejestr systemu PEAR, 247
  - sesja, 341
  - tłumaczeń, 418
  - tryby dostępu, 230

- tymczasowa nazwa, 304
- uchwyt, 230, 231
- wczytywanie całej zawartości, 236
- wskaźnik, 626
- zamykanie, 231
- zewnątrzny, 53
- pobieranie listy właściwości obiektu, 159
- pobieranie listy zadeklarowanych klas, 158
- podprogramy
  - architektura, 574
  - bezpieczeństwo, 574
  - handlers, 588
  - modyfikowanie, 586
  - obsługi, 588
  - składnia, 573
  - składowane, 573, 574, 575, 579, 580, 583, 586, 589, 590, 601
  - spójność, 574
  - status, 587
  - usuwanie, 586
  - wydajność, 574
  - wykonywanie, 573, 579
  - wyświetlenie polecenia, 588
  - wywoływanie z innego podprogramu, 586
  - zarządzanie, 573, 580
- podzapytanie, 613, 623, 624, 625
- samołączenie, 623
- self-join, 623
- stosowanie, 625
- subquiere, 623
- polecenie
  - bazy danych, 468, 470
  - DESCRIBE VIEW, 607
  - du, 227
  - GRANT, 511
  - if, 289
  - LOAD DATA INFILE, 655
  - prepared statements, 537
  - przygotowane, 537, 565
  - REVOKE, 511
  - rm, 276
  - SELECT INTO OUTFILE, 661, 662
  - SET PASSWORD, 512
  - SHOW CREATE VIEW, 608
  - SHOW TRIGGERS, 596
  - SQL, 537, 563, 564
  - systemowe, 227, 239, 241, 242, 243, 244
  - USE, 468, 474
- polimorfizm, 141, 142, 143, 162
- polskie znaki diakrytyczne, 205
- połączenie
  - bezpieczne, 511
  - używające gniazd, 314
  - z usługą rejestrującą, 179, 180
- POP3, 32, 292, 313
- port, 380
  - numer, 313, 314
  - komunikacyjny, 313
  - skaner, 315, 322
  - SSH, 512
- POSIX, 189, 190, 192, 195
  - biblioteka, 32
  - Portale Operating System Interface for Unix, 190
- postdekrementacja, 88
- PostgreSQL, 558
- postinkrementacja, 88
- poszukiwanie wzorca, 198
- powrót karetki, 317
- poziom zgłaszania błędów, 176
- predekrementacja, 88
- preinkrementacja, 88
- procedura
  - składowana, 577
- program
  - mysql, 467
  - mysqldadmin, 467, 475, 476, 477, 480
  - mysqlimport, 653, 654
  - tar, 454
- programowanie
  - obiektowe, 33, 51, 72, 141, 142, 143, 146, 153, 160, 161, 394, 537
  - proceduralne, 33, 537
- protokół, 32
  - bezzstanowy, 339
  - HTTP, 285, 286, 288, 299
  - HTTP/1.1, 48
  - Secure Socket Layer, 286
  - SSL, 286
  - Transport Layer Security, 327
- prototyp, 67
- przepisywanie adresów URL
  - URL rewriting, 340
- przestrzeń nazw, 27, 30, 161, 172, 173
- przeszukiwanie
  - baz danych, 631
  - bez uwzględniania wielkości liter, 193
  - z uwzględnieniem wielkości liter, 192
- przetłumaczenie tekstów, 418
- przetwarzanie łańcuchów znaków, 33
- przewinięcie wiersza, 317
- Pyrus, 245, 253
  - instalacje menedżera, 253
- Python, 30, 33, 175, 181, 309

## R

- RADIUS, 292
- raportowanie
  - błędów, 49
- Raymond Eric S., 33

- RDN
    - Relative Distinguished Name, 327
  - Red Hat, 441, 442
  - RedHat Directory Server, 325
  - referencja, 49, 77, 79, 126, 161, 162, 176
    - tworzenie, 77
  - Resig John, 405
  - rejestr systemowy, 248
  - rejestracja
    - błędu, 178
    - funkcja, 179
    - identyfikator, 179
    - opcje, 180
    - użytkowników, 485
  - rekord
    - baza danych, 485
    - MX, 313
  - rekurencja, 111, 127
  - replikacja, 538
  - repozytorium, 416, 417
  - root
    - hasło użytkownika, 512
    - uprawnienia użytkownika, 468
  - rozszerzenie
    - .html, 39, 40, 357
    - .json, 39, 40
    - .php, 39, 40, 43, 62
    - .php5, 39, 40
    - .phtml, 62
    - Filter, 275, 277, 278
  - RSA Security Inc, 290
  - RSS, 379, 381, 382, 383, 384, 386
    - agregator, 381
    - czytnik, 381
    - Really Simple Syndication, 379, 381
  - rzutowanie, 73
    - operator, 73
- S**
- SAMBA, 292
  - schemat adresowania IP, 309, 310
  - sekcja
    - Global Environment, 40
    - środowisko ogólne, 40
  - sekretny kod, 94
  - sekwencje sterujące, 91, 93
  - sendmail, 33, 316, 317
  - separacja, 357
  - serializacja, 49
  - serwer
    - administrator, 39
    - Apache, 35, 36, 37, 38, 39, 40, 41, 43, 44, 45, 51, 52, 53, 62, 285, 286, 287, 292, 395, 396, 397, 398, 399
    - Apache2, 39
    - asynchroniczna komunikacja, 403
    - baz danych, 61, 441, 442, 467
    - binarne wersje, 36
    - CVS, 37
    - dedykowany, 61
    - DNS, 309, 310
    - dokumentacja, 37
    - IIS, 29, 35, 36, 37, 41
    - IIS 7, 41
    - instalacja, 35, 40
    - kod źródłowy, 36, 451
    - komunikacja z aplikacją, 404
    - konfiguracja, 394, 398, 468
    - konta użytkowników, 512
    - kopia, 36
    - LDAP, 326
    - mirror, 36
    - MySQL, 441, 443, 444, 445, 446, 448, 449, 511, 512, 513, 514, 515, 516, 518, 519, 520, 523, 524, 526, 528, 529, 530, 531, 534, 535, 555, 560, 561, 564, 601, 602, 611, 645, 646, 653, 655, 659
    - nazw domen, 310
    - obsługa SSL, 36
    - obszary chronione, 51
    - Oracle, 555
    - pamięć operacyjna, 486
    - pobieranie, 36
    - podpis, 53
    - proces, 511
    - przerwanie sesji, 52
    - przesyłanie plików, 299, 300, 304, 305, 306
    - środowisko, 240
    - testowanie połączenia, 321
    - wielkość pliku przesyłanego, 301
    - wirtualny, 61
    - współdzielony, 61, 179
    - WWW, 35, 36, 42, 44, 52, 53, 60, 61, 62, 80, 82, 177, 178, 239, 240, 241, 286, 289, 302, 305, 357, 360, 393, 394, 512, 659
    - Xitami, 37
    - zabezpieczenia procesu, 511
  - serwer bazy danych
    - SQLite, 30
  - sesja
    - obsługa, 339
    - proces obsługi, 340
    - ścieżka do pliku, 341
  - SHA1, 29
  - sieć, 309, 314, 324
    - czynności, 309
    - konfiguracja, 322
    - popularne operacje, 316, 321
    - testowanie przepustowości łącza, 324
    - usługi, 379
  - SimpleXML, 30, 379, 387, 389, 390, 391

- sklep internetowy, 423
- składnia
  - heredoc, 93
  - języka programowania, 358
  - nowdoc, 93
  - skryptów powłoki, 66
- skrypt
  - aktualizacja, 416
  - mysql\_install\_db, 454, 455
  - powłoki, 63, 66, 67
- słownik
  - cracklib\_dict.pwd, 296
- słowo ignorowane, 637
  - mechanizm, 637
  - stopwords, 637
- słowo kluczowe
  - \$this, 143, 145, 148, 149, 150, 154, 157
  - abstract, 171
  - clone, 162
  - final, 147
  - implements, 169
  - instanceof, 157, 158
  - parent, 155
  - protected, 146
  - public, 145, 150
  - self, 157, 168
  - static, 168
  - var, 176
- Smarty, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377
  - bezpieczeństwo, 359
  - kompilacja szablonów, 359
  - konfiguracja i rozszerzenia, 359
  - logika prezentacji, 359
  - przedstawienie mechanizmu, 359
  - wykorzystanie pamięci podręcznej, 359
- SMTP, 316, 317, 512
- sortowanie, 128, 613, 618
  - bąbelkowe, 128
  - dane, 442
  - heapsort, 128
  - kluczy w odwrotnej kolejności, 132
  - kolejność numeryczna, 129
  - kolejność rosnąca, 130, 132
  - na podstawie kryteriów określonych przez użytkownika, 132
  - na podstawie wartości ich kodów ASCII, 129
  - na podstawie wartości kluczy, 132
  - odwrotna (malejąca) kolejność, 130
  - przez kopcowanie, 128
  - quicksort, 128
  - SORT\_LOCALE\_STRING, 128
  - SORT\_NUMERIC, 128, 129
  - SORT\_REGULAR, 128, 129
  - SORT\_STRING, 128, 129
  - Stella, 128
  - szybkie, 128
  - w kolejności malejące z zachowaniem odwzorowania kluczy i wartości, 131
  - w porządku naturalnym, 131
  - wartości liczbowych, 128
- specyfikator typu, 69
- SPL, 186, 187
  - Standardowa Biblioteka PHP, 186
- sprawdzanie typu klasy pochodnej obiektu, 159
- sprawdzanie, czy istnieje metoda, 159
- SQL, 291, 297, 467, 468, 470, 471, 474, 476, 477, 482
  - sekwencja zapytań, 601
  - zarządzanie kodem, 573
- SQLite, 341
  - obsługa, 30
- SQLite 3.X, 558
- SSH., 313
- SSL, 511, 526, 531, 532, 533, 534
- stała, 84
  - definiowanie, 85
- static, 80
- statyczne wiązanie, 30
- stos
  - last in, first out, 121
  - LIFO, 121
- struktura
  - Appliance, 72, 73
  - break, 63
  - continue, 63
  - declare, 63
  - foreach, 63, 98
  - if-else-elseif, 63
  - include, 63, 101, 102
  - require, 63, 103
  - require\_once, 103
  - sterująca, 63, 93, 94
  - while, 63, 96
- Supraski Zeev, 28
- Sun Microsystems, 441
- Swatch, 256
- symbol znakowy, 206
- system
  - budowania, 38
  - operacyjny, 223, 240, 241, 242, 243, 393
  - operacyjny poprawki, 512
- szkielet
  - CakePHP, 426
  - Ruby on Rails, 425
  - Solar, 426
- szkielet aplikacji internetowej, 423
  - web frameworks, 423
- szyfrowanie, 287, 289

## Ś

ścieżka

- bezwzględna, 225
- dodatkowe informacje, 224

## T

tabela, 297, 485, 601, 606, 662

- baza danych, 485
- definicja, 633
- docelowa, 657, 658
- dodanie danych, 648
- przestrzeń, 647
- struktura, 633
- table space, 647
- tworzenie, 646
- użytkowników, 291

tablica, 72, 115,

- \$\_COOKIE, 83
- \$\_ENV, 84
- \$\_FILES, 83, 299, 300, 302
- \$\_GET, 82
- \$\_POST, 82, 83, 272
- \$\_REQUEST, 84
- \$\_SERVER, 82
- \$\_SESSION, 84
- \$GLOBALS, 79, 84
- asocjacyjna, 117, 120, 123, 127, 129, 137, 138, 158, 159, 177, 224, 259

definicja, 115

dodawanie elementów, 121

dzielenie, 134, 140

elementy, 115, 118, 127, 128, 129, 134, 140

indeks, 72, 198

indeksowana liczbami, 117, 123

klucz, 72, 115, 128, 134, 307

losowa zmiana kolejności elementów, 139

łączenie, 115, 134, 135

łączenie rekurencyjne, 134

odczytywanie zawartości pliku, 231

odnajdywanie elementów, 122

odzworowanie kluczy i wartości, 130

określanie

różnicy, 137, 138

wielkości, 127

pobieranie

aktualnego klucza i wartości, 125

fragmentu, 135

klucza, 123

klucza aktualnego elementu, 124

wartości, 123

pobieranie wartości aktualnego elementu, 124

poszukiwanie wartości, 123

powtórzenia, 128

predefiniowana, 82

przeglądanie, 115, 124

przekazywanie wartości do funkcji, 126

przesuwanie wskaźnika, 125

przeszukiwanie, 122, 197

sortowanie, 115, 128, 129

sortowanie kluczy, 132

sumowanie wartości, 139

tworzenie, 115, 116, 117

unikalne wartości, 128

unikalność, 127

usuwanie elementów, 121

wartości, 116, 117, 124, 126, 128, 130, 132, 137, 138

wczytanie zawartości katalogu, 225, 231, 232, 238, 239

wczytywanie pliku CSV, 233

wejściowa, 128, 129, 136

wielkość, 117, 127

wielowymiarowa, 72, 116, 119, 127, 140

wskaźnik, 116, 626

wycinanie elementów, 136

wyświetlanie, 120

zawartości, 119

wyznaczanie części wspólnej, 136

zamiana kluczy i wartości, 129

zapis do pliku, 290

zapis do zmiennej, 259

zawierająca tablice, 116

zmiana kolejności elementów, 129, 139

zwracanie losowego zbioru kluczy, 139

TeX DataKonsult AB, 441

teoria informacji, 70

testowanie instalacji, 35, 39, 41

TLS, 327

połączenie, 327

Transport Layer Security, 327

tłumaczenie

tekstów, 420

wyświetlanie daty, liczby oraz godziny, 420

transakcja, 485, 538, 645, 649, 650

atomicity, 646

committed, 485, 645

consistency, 646

durability, 646

isolation, 646

izolacja, 646

niepodzielność, 646

PDO, 570, 571

realizacja, 648

rolled back, 485, 645

spójność, 646

trwałość, 646

wycofanie, 485, 645

zatwierdzenie, 485, 645

TripleDES, 29

try/catch, 30

## tryb

- bezpieczny, 50, 394, 395
- interaktywny, 467, 468, 470, 471, 474, 475
- safe mode, 50, 394
- wsadowy, 467

## typ danych, 67, 70, 71, 73, 74

- automatyczna konwersja, 74, 85
- konwersja, 73, 75
- liczba całkowita, 70, 71
- łańcuch znaków, 69, 70, 71, 73, 74, 84, 91
- MySQL, 485
- obiekt, 72, 74, 162
- rzutowanie, 73
- skalarny, 70, 92, 109
- tablice, 70, 71, 72, 73
- wartość logiczna, 70
- złożony, 71, 72
- zmiennie, 74
- zmiennoprzecinkowa, 70, 71

## U

Unicode, 27, 30, 415

Unix, 63, 248, 255, 256, 295

uprawnienia, 576

uproszczona składnia, 65

## URL

- przepisywanie adresów, 62
- rewriting, 62

## usługa, 314

- DNS, 309, 310
- katalogowe, 325
- pobieranie nazwy, 314
- sieciowa, 30, 379, 380, 381, 423
- systemowa, 512
- web services, 379
- wyłączenie, 512

## UTC, 255

uwierzytelnianie, 285, 287, 289, 290, 291

baza danych, 291, 292, 293

drzewo katalogów, 289

HTTP, 286

kod 401, 288

metodologia, 288

nazwa użytkownika i hasło, 285, 286, 287, 288, 289

niezaszyfrowane informacje, 286

plik, 289

procedura, 285

stałe dane, 289

użytkowników, 32, 285, 286, 287, 288, 290

w oparciu o protokół HTTP, 285

zarządzanie, 294

zmiennie PHP, 285, 287

## V

Validate\_PL, 189

Validate\_US, 189, 219, 220, 221

instalacja, 220

stosowanie, 220

## W

Walk Jog Run, 379

## warstwa

logiki biznesowej, 357

prezentacji, 357, 362, 363

## warstwa abstrakcji

dostęp do danych, 32

funkcje, 32

Java Database Connectivity, 556

JDBC, 556

MDB2, 556

ODBC, 556

Open Database Connectivity, 556

PDO, 556

Perl Database Interface, 556

Perl DBI, 556

## wartość

&amp;amp;, 53

0, 70

array, 75

atrybut obiektu, 74

boolean, 73, 75

double, 49, 71, 73, 75

dziesiętna, 70

float, 49, 69, 71

integer, 69, 73, 75

LD\_LIBRARY\_PATH, 51

liczba całkowita, 73, 153

liczba rzeczywista, 71

logiczna, 70

łańcuch znaków, 73

monetarna, 70, 71

NULL, 48, 49, 50, 51, 56, 57, 58, 59

object, 73, 74, 75

PHP\_, 50

resource, 75

string, 68, 69, 70, 73, 75

text/html, 56

unknown type, 75

zależności, 89

zmiennoprzecinkowa, 69, 73, 74, 153

## wartość dyrektywy

php\_admin\_flag, 45, 46

php\_admin\_value, 45

php\_flag, 45

php\_value, 45

- weryfikacja
    - danych, 277
  - widok, 425, 433
    - aktualizujące, 609
    - korzystanie w serwerze, 602
    - modyfikacja, 609
    - postać wyników, 604
    - przedstawienie, 602
    - przekazanie parametrów, 605
    - stosowanie, 610
    - usuwanie, 609
    - view, 601
    - wyświetlanie informacji, 607
  - wiersz
    - baza danych, 485
    - kodu, 31
    - komunikatu, 178
  - Windows,
    - 7, 41
    - Server 2003, 41
    - Server 2008, 41
    - Vista, 41
    - XP, 41
  - Windows-1251, 419
  - wirtualizacja, 61
  - właściwości
    - nazwy, 158
    - pobieranie nazw, 158
    - statyczne, 157, 168
  - WML, 56
    - Wireless Markup Language, 56
  - Woozor, 379
  - WordPress, 62
  - wskaźnik
    - pointer, 626
  - wyjątek, 30, 175, 181, 182, 183, 184, 185, 186, 187
    - implementacja obsługi, 182
    - klasy, 186, 187
    - konstruktor domyślny, 182
    - konstruktor przeciążony, 183
    - metody, 183
    - obsługa, 175, 181, 182
    - predefiniowany, 186
    - przechwytywanie wielu, 185
    - SPL, 186
  - wynik
    - cursor, 626
    - kursor, 626
    - zbiór, 626
  - wyrażenia, 85
  - wyrażenie regularne, 189, 202, 210, 211
    - funkcje obsługi, 192
    - metaznaki, 196
    - modyfikatory, 196
    - składnia, 190, 195
  - WYSIWYG, 59
    - what you see is what you get, 59
  - wyszukiwanie, 631
    - pełnotekstowe, 642
    - pełnotekstowe z elementami logicznymi, 638
    - z użyciem formularzy HTML, 631, 639
  - wyświetlanie
    - numerów stron, 621
  - wywołanie
    - funkcji, 31
    - kolejność, 31
  - wyzwalacze, 591, 592, 594, 598, 599
    - konwencja nazewnictwa, 595
    - modyfikacja, 598
    - prezentacja, 591
    - rejestracja danych, 592
    - trigger, 591, 592, 597
    - usuwanie, 598
    - weryfikacja, 592
    - wykonywanie, 593
    - wymuszanie integralności odwołań, 592
    - wyświetlanie, 596
  - wzorzec
    - MVC, 423
- ## X
- XAMPP, 36
  - Xerox Corporation, 299
  - XHTML, 614
  - XML, 30, 44, 47, 59, 379, 381, 384, 387, 388, 389, 390, 391, 403, 404
    - asynchroniczne, 403
- ## Z
- zabezpieczanie, 273
    - aplikacji, 54, 394
    - argument wywoływanych programów, 275
    - danych, 275, 277, 278
    - mechanizmy, 537
    - metaznak wiersza poleceń, 276
    - serwer MySQL, 511
    - systemy, 511
    - witryn WWW, 393
  - Zandstra Matt, 141
  - zapis
    - dziesiętny, 71
    - ósemkowy, 71
    - szesnastkowy, 71
  - zaporą ogniową, 313, 380, 512
    - uszczelnianie, 512
  - zapytanie
    - kursor, 613, 626, 627, 628
    - optymalizacja zapytań, 631
    - podzapytania, 613



- zapytanie
  - podział wyników na strony, 613, 619
  - prezentacja danych w formacie tabelarycznym, 613
  - query expansion, 636
  - POSIX, 33
  - rozwijanie, 636
  - sortowanie danych w formacie tabelarycznym, 613
  - subquery, 613
  - unikalność, 631
- zarządzanie
  - konta użytkowników, 511
  - uprawnienia, 601
- zasięg
  - chroniony, 146, 151
  - prywatny, 146, 151
  - publiczny, 145, 150
  - właściwości statyczne, 168
- zasięg dyrektywy, 46
  - PHP\_INI\_ALL, 46
  - PHP\_INI\_PERDIR, 46, 57
  - PHP\_INI\_SYSTEM, 46, 48, 49, 50, 51, 52, 54, 55, 56, 57, 58
  - PHP\_INI\_USER, 46
- zasób 231
  - obsługa, 229
- zastępowanie tekstu
  - bez uwzględniania wielkości liter, 194
  - z uwzględnieniem wielkości liter, 193
- zastępowanie wszystkich wystąpień wzorca, 199
- zbiór znaków, 419
- zdarzenie
  - asynchroniczne, 404
- Zend Framework, 423, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 438, 439
  - mechanizm skryptowy, 28
  - szkielet, 423
- Zend Studio, 60
- Zend Technologies Ltd, 28, 37, 41, 60
- zip dla systemu Windows
  - zip package, 37
- zmienna, 66, 76, 93, 108
  - \$\_SERVER, 273
  - \$\_SERVER['PHP\_AUTH\_PW'], 285, 287, 288, 289
  - \$\_SERVER['PHP\_AUTH\_USER'], 285, 287, 288, 289
  - \$category, 96
  - \$count, 96
  - \$date, 65
  - \$HTTP\_RAW\_POST, 57
  - \$somevar, 79
  - automatyczne usuwanie, 32
  - COOKIE, 54, 55
  - deklaracje, 76, 578
  - dostępna globalnie, 54
  - ENVIRONMENT, 54, 55
  - GET, 53, 54, 55, 56
  - globalna, 78, 79
  - lokalna, 78, 79, 80
  - łańcuch znaków, 71
  - parametry funkcji, 78
  - PHP\_INCLUDE\_PATH, 57
  - pole formularza, 280
  - położenie deklaracji, 78
  - POST, 53, 54, 55, 56, 57
  - przekształcenia, 79
  - przypisanie przez referencję, 77
  - przypisywanie wartości, 77
  - register\_globals, 54, 55
  - rzutowanie, 32
  - SERVER, 54, 55
  - sessionid, 54
  - statyczna, 78, 80
  - superglobalna, 54, 80
  - symboliczna, 28
  - ustawianie, 578
  - wartość, 78, 79, 84, 88, 89
  - zasięg, 78
  - zakres, 30
  - zawierająca tablicę, 119
  - zewnętrzna, 53, 54
  - zmienna, 84
  - zwyczajna, 81
- Zmievski Andrei, 359
- znacznik
  - %>, 65
  - ?>, 43
  - <% , 65
  - <?php, 43
  - <script>, 65
  - czasu, 255, 256, 260, 261
  - czasu Uniksowy, 255, 257, 258, 259, 260, 261
  - HTML, 404
  - krótki, 46, 47, 64, 65
  - krótki otwierający, 47
  - otwierający, 64
  - short open tags, 47
  - short tags, 46
  - timestamps, 255
  - usuwanie, 234, 277
  - zamykający, 64
- znak
  - \$, 76, 84, 85, 92, 145, 147, 192
  - &, 107, 208
  - :, 94
  - ~, 244
  - <<<, 93
  - cudzysłów, 92
  - endfor, 94
  - endforeach, 94
  - endif, 94
  - end-of-file, 229
  - endswitch, 94

endwhile, 94  
EOF, 229, 234, 235, 237  
escape character, 78  
klasy, 192  
konwersja, 276  
końca wiersza, 59  
końca pliku, 229  
liczba znaków, 234  
łańcuch, 68, 86  
\n, 91  
nowego wiersza, 92, 231, 235  
odwrotnego ukośnika, 75, 78, 92, 93, 191, 192, 199  
oznaczanie potencjalnie niebezpiecznych, 242  
powrót karetki, 92  
predefiniowana lista, 210  
predefiniowane zakresy, 192  
rozpoznawanie końca pliku, 229  
rozpoznawanie nowego wiersza, 229  
specjalny, 199, 206, 207  
tabulacja pozioma, 92  
tyldy, 244  
unikowy, 78  
w zapisie ósemkowym, 92

# PHP i MySQL. Od podstaw

PHP i MySQL to duet, na którym opiera się ogromna większość trochę bardziej zaawansowanych stron w sieci. Czemu zawdzięcza on swą popularność? Niezwykłemu dopasowaniu do potrzeb, łatwej konfiguracji oraz ogromnej społeczności, zawsze chętniej do pomocy. Kiedy zaprzęgniesz ten zestaw do pracy, już po kilku godzinach zobaczysz jej pierwsze efekty!

Co zrobić, żeby osiągnąć jeszcze więcej? Nic prostszego! Sięgnij po tę książkę i zacznij swoją przygodę z PHP oraz MySQL. Na samym początku dowiesz się, jak przygotować środowisko pracy, oraz poznasz podstawy programowania w języku PHP. Potem płynnie przejdziesz do zdobywania wiedzy na temat programowania obiektowego, wyrażań regularnych, obsługi formularzy HTML oraz integracji z usługami katalogowymi LDAP. Nauczysz się wykorzystywać mechanizm sesji, technologię AJAX oraz Zend Framework. Po przeprowadzeniu Cię przez komplet informacji na temat PHP autor przedstawi Ci bazę danych MySQL. Poznasz jej mocne elementy, sposoby podłączania klientów oraz podstawy administracji. Może w to nie wierzysz, ale po przeczytaniu tej książki nawet skomplikowane mechanizmy bazy MySQL nie będą Ci obce! Na sam koniec odkryjesz, jak wiedzę na temat PHP oraz MySQL połączyć w całość i zrealizować Twój nawet najbardziej ambitny projekt! Książka ta jest obowiązkową pozycją na półce każdego entuzjasty tworzenia dynamicznych stron WWW!

Sprawdź, jak wykorzystać możliwości PHP i MySQL w zakresie projektowania stron internetowych!

- ▼ Przygotowanie środowiska do pracy
- ▼ Podstawy PHP – typy danych, zmienne, funkcje, tablice
- ▼ Programowanie obiektowe w PHP
- ▼ Obsługa wyjątków i błędów
- ▼ Operacje na łańcuchach znaków, wyrażenia regularne
- ▼ Obsługa plików i korzystanie z funkcji systemu operacyjnego
- ▼ Wykorzystanie pakietów PEAR
- ▼ Operacje na dacie i czasie
- ▼ Obsługa formularzy HTML
- ▼ Uwierzytelnianie użytkowników
- ▼ Integracja PHP z LDAP
- ▼ Zarządzanie sesją
- ▼ Szablony stron – pakiet Smarty
- ▼ Zabezpieczanie witryn WWW
- ▼ Zastosowanie technologii AJAX z jQuery
- ▼ Wykorzystanie Zend Framework
- ▼ Możliwości bazy danych MySQL
- ▼ Instalacja i konfiguracja serwera MySQL
- ▼ Mechanizmy składowania i typy danych w MySQL
- ▼ Zabezpieczanie serwerów MySQL
- ▼ Pobieranie danych z bazy i operacje na nich w PHP
- ▼ Wykorzystanie widoków
- ▼ Mechanizm transakcji



**Helion**

Sprawdź najnowsze promocje:

📌 <http://helion.pl/promocje>

📌 Książki najchętniej czytane:

📌 <http://helion.pl/bestsellery>

Zamów informacje o nowościach:

📌 <http://helion.pl/nowości>

**Helion SA**

ul. Kosciuszki 1c, 44-100 Gliwice

tel.: 32 230 98 63

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

<http://helion.pl>

Nr katalogowy: 6211



Księgarnia internetowa:

<http://helion.pl>



Zamówienia telefoniczne:

**0 801 339900**



**0 601 339900**

**helion.pl**  
księgarnia  
internetowa

**Cena 109,00 zł**

ISBN 978-83-246-3056-1



9 788324 630561

Informatyka w najlepszym wydaniu