

Tytuł oryginału: PHP and MySQL® Web Development, Fourth Edition

Tłumaczenie: Daniel Kaczmarek

Projekt okładki: Maciek Pasek

ISBN: 978-83-246-3177-3

© Helion S.A. 2009

Authorized translation from the English language edition, entitled: PHP and MySQL Web Development, Fourth Edition, ISBN 0672329166, by Luke Welling and Laura Thomson, published by Pearson Education, Inc., publishing as Sams Publishing, Copyright © 2009 by Pearson Education, Inc.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education Inc.

Polish language edition published by Helion S.A.

Copyright © 2009

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

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

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

Pliki z przykładami omawianymi w książce można znaleźć pod adresem:

<ftp://ftp.helion.pl/przyklady/phmsv4.zip>

Materiały graficzne na okładce zostały wykorzystane za zgodą iStockPhoto Inc.

Wydawnictwo HELION

ul. Kościuszki 1c, 44-100 GLIWICE

tel. 032 231 22 19, 032 230 98 63

e-mail: helion@helion.pl

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

Drogi Czytelniku!

Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres

<http://helion.pl/user/opinie/phms4v>

Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

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

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

Spis treści

	O autorach	23
	O współautorach	25
	Wprowadzenie	27
Część I	Stosowanie PHP	37
Rozdział 1.	Podstawowy kurs PHP	39
	Zastosowanie PHP	40
	Tworzenie przykładowej aplikacji: „Części samochodowe Janka”	40
	Formularz zamówienia	40
	Przetwarzanie formularza	41
	Osadzanie PHP w HTML	42
	Zastosowanie znaczników PHP	43
	Instrukcje PHP	44
	Odstępy	44
	Komentarze	45
	Dodawanie zawartości dynamicznej	45
	Wywoływanie funkcji	46
	Używanie funkcji date()	46
	Dostęp do zmiennych formularza	47
	Zmienne formularza	47
	Łączenie ciągów	49
	Zmienne i ciągi znaków	50
	Identyfikatory	51
	Typy zmiennych	51
	Typy danych w PHP	51
	Siła typu	52
	Rzutowanie typu	52
	Zmienne zmiennych	53
	Deklarowanie i używanie stałych	53
	Zasięg zmiennych	54
	Używanie operatorów	55
	Operatory arytmetyczne	55
	Operatory ciągów	56
	Operatory przypisania	56
	Operatory porównań	58
	Operatory logiczne	59
	Operatory bitowe	60
	Pozostałe operatory	60

Obliczanie sum w formularzu	62
Pierwszeństwo i kolejność	63
Zarządzanie zmiennymi	65
Sprawdzanie i ustawianie typów zmiennych	65
Sprawdzanie stanu zmiennej	66
Reinterpretacja zmiennych	67
Podjmowanie decyzji za pomocą instrukcji warunkowych	67
Instrukcja if	67
Bloki kodu	68
Instrukcja else	68
Instrukcja elseif	69
Instrukcja switch	69
Porównanie różnych instrukcji warunkowych	71
Powtarzanie działań przy użyciu iteracji	71
Pętla while	72
Pętla for i foreach	73
Pętla do..while	74
Wyłamywanie się ze struktury skryptu	75
Używanie alternatywnych składni struktur sterujących	75
Używanie struktury declare	76
W następnym rozdziale	76
Rozdział 2. Przechowywanie i wyszukiwanie danych	77
Zapisywanie danych do późniejszego użycia	77
Przechowywanie i wyszukiwanie zamówień Janka	78
Przetwarzanie plików	79
Otwieranie pliku	79
Tryby otwarcia pliku	79
Stosowanie funkcji fopen() do otwarcia pliku	80
Otwieranie pliku przez protokół FTP lub HTTP	82
Problemy z otwieraniem plików	82
Zapisywanie danych w pliku	84
Parametry funkcji fwrite()	85
Formaty plików	85
Zamykanie pliku	86
Odczyt z pliku	87
Otwieranie pliku w celu odczytu — fopen()	89
Wiedzieć, kiedy przestać — feof()	89
Odczytywanie pliku wiersz po wierszu — fgets(), fgets() i fgetsv()	89
Odczyt całego pliku — readfile(), fpassthru(), file()	90
Odczyt pojedynczego znaku — fgetc()	91
Odczytywanie zadanej długości — fread()	91
Inne przydatne funkcje plikowe	91
Sprawdzanie istnienia pliku — file_exists()	92
Określanie wielkości pliku — filesize()	92
Kasowanie pliku — unlink()	92
Poruszanie się wewnątrz pliku — rewind(), fseek() i ftell()	92
Blokowanie pliku	93
Lepszy sposób obróbki danych — systemy zarządzania bazami danych	94
Problemy związane ze stosowaniem plików jednorodnych	95
Jak RDBMS rozwiązują powyższe problemy?	95
Propozycje dalszych lektur	96
W następnym rozdziale	96

Rozdział 3.	Stosowanie tablic	97
	Czym są tablice?	97
	Tablice indeksowane numerycznie	98
	Inicjowanie tablic indeksowanych numerycznie	98
	Dostęp do zawartości tablicy	99
	Dostęp do tablic przy zastosowaniu pętli	100
	Tablice z innymi indeksami	100
	Inicjowanie tablicy	100
	Dostęp do elementów tablicy	101
	Stosowanie pętli	101
	Operatory tablicowe	103
	Tablice wielowymiarowe	103
	Sortowanie tablic	106
	Stosowanie funkcji sort()	106
	Stosowanie funkcji asort() i ksort() do porządkowania tablic	107
	Sortowanie odwrotne	107
	Sortowanie tablic wielowymiarowych	108
	Typy sortowań definiowane przez użytkownika	108
	Odwrotne sortowanie zdefiniowane przez użytkownika	109
	Zmiany kolejności elementów w tablicach	110
	Stosowanie funkcji shuffle()	110
	Stosowanie funkcji array_reverse()	111
	Ładowanie tablic z plików	112
	Wykonywanie innych działań na tablicach	114
	Poruszanie się wewnątrz tablicy	
	— funkcje each(), current(), reset(), end(), next(), pos() i prev()	114
	Dołączanie dowolnej funkcji do każdego elementu tablicy — funkcja array_walk()	115
	Liczenie elementów tablicy: count(), sizeof() i array_count_values()	116
	Konwersja tablic na zmienne skalarne — funkcja extract()	117
	Propozycje dalszych lektur	118
	W następnym rozdziale	118
Rozdział 4.	Manipulowanie ciągami i wyrażenia regularne	119
	Przykładowa aplikacja — Inteligentny Formularz Pocztowy	119
	Formatowanie ciągów	121
	Przycinanie ciągów — funkcje chop(), ltrim() i trim()	121
	Formatowanie ciągów w celu ich prezentacji	122
	Formatowanie ciągów do przechowania — funkcje addslashes() i stripslashes()	125
	Łączenie i rozdzielanie ciągów za pomocą funkcji ciągów	127
	Stosowanie funkcji explode(), implode() i join()	127
	Stosowanie funkcji strtok()	128
	Stosowanie funkcji substr()	128
	Porównywanie ciągów	129
	Porządkowanie ciągów — funkcje strcmp(), strcasecmp() i strnatcmp()	129
	Sprawdzanie długości ciągu za pomocą funkcji strlen()	130
	Dopasowywanie i zamiana podciągów za pomocą funkcji ciągów	130
	Znajdowanie ciągów w ciągach — funkcje strstr(), strchr(), strrchr() i stristr()	131
	Odnajdywanie pozycji podciągu — funkcje strpos() i strrpos()	131
	Zamiana podciągów — funkcje str_replace() i substr_replace()	132
	Wprowadzenie do wyrażen regularnych	133
	Podstawy	133
	Zbiory i klasy znaków	134
	Powtarzalność	135
	Podwyrażenia	135
	Podwyrażenia policzalne	135

	Kotwiczenie na początku lub na końcu ciągu	136
	Rozgałęzianie	136
	Dopasowywanie specjalnych znaków literowych	136
	Podsumowanie znaków specjalnych	137
	Umieszczanie wszystkiego razem (Inteligentny Formularz)	137
	Odnajdywanie podciągów za pomocą wyrażeń regularnych	138
	Zamiana podciągów za pomocą wyrażeń regularnych	139
	Rozdzielanie ciągów za pomocą wyrażeń regularnych	139
	Propozycje dalszych lektur	140
	W następnym rozdziale	140
Rozdział 5.	Ponowne wykorzystanie kodu i tworzenie funkcji	141
	Zalety ponownego stosowania kodu	141
	Koszt	142
	Niezawodność	142
	Spójność	142
	Stosowanie funkcji require() i include()	142
	Rozszerzenia plików i require()	143
	Stosowanie require() w szablonach stron WWW	144
	Stosowanie opcji auto_prepend_file i auto_append_file	148
	Stosowanie funkcji w PHP	149
	Wywoływanie funkcji	149
	Wywołanie niezdefiniowanej funkcji	151
	Wielkość liter a nazwy funkcji	152
	Definiowanie własnych funkcji	152
	Podstawowa struktura funkcji	152
	Nadawanie nazwy funkcji	153
	Parametry	154
	Zasięg	156
	Przekazanie przez referencję czy przekazanie przez wartość?	158
	Stosowanie słowa kluczowego return	159
	Zwracanie wartości przez funkcje	160
	Implementacja rekurencji	161
	Przestrzenie nazw	162
	Propozycje dalszych lektur	163
	W następnym rozdziale	163
Rozdział 6.	Obiektowy PHP	165
	Koncepcje programowania obiektowego	165
	Klasy i obiekty	166
	Polimorfizm	167
	Dziedziczenie	167
	Tworzenie klas, atrybutów i operacji w PHP	168
	Struktura klasy	168
	Konstruktory	168
	Destruktory	169
	Tworzenie egzemplarzy	169
	Stosowanie atrybutów klasy	170
	Kontrolowanie dostępu przy użyciu private i public	172
	Wywoływanie operacji klas	172
	Implementacja dziedziczenia w PHP	173
	Kontrolowanie widoczności w trakcie dziedziczenia przy użyciu private i protected	174
	Unieważnianie	175
	Zapobieganie dziedziczeniu i unieważnianiu przy użyciu final	176
	Wielodziedziczenie	177
	Implementowanie interfejsów	177

Tworzenie klas	178
Tworzenie kodu dla własnej klasy	179
Zaawansowane mechanizmy obiektowe w PHP	186
Używanie stałych klasowych	186
Implementowanie metod statycznych	186
Sprawdzanie typu klasy i wskazywanie typu	186
Późne wiązania statyczne	187
Klonowanie obiektów	188
Używanie klas abstrakcyjnych	188
Przeciążanie metod przy użyciu <code>__call()</code>	188
Używanie metody <code>__autoload()</code>	189
Implementowanie iteratorów i iteracji	190
Przekształcanie klas w łańcuchy znaków	191
Używanie API Reflection	192
W następnym rozdziale	192
Rozdział 7. Obsługa błędów i wyjątków	195
Koncepcja obsługi wyjątków	195
Klasa Exception	196
Wyjątki definiowane przez użytkownika	197
Wyjątki w Częściach samochodowych Janka	200
Wyjątki i inne mechanizmy obsługi błędów w PHP	202
Propozycje dalszych lektur	203
W następnym rozdziale	203
Część II Stosowanie MySQL	205
Rozdział 8. Projektowanie internetowej bazy danych	207
Koncepcje relacyjnych baz danych	208
Tabele	208
Kolumny	208
Wiersze	208
Wartości	208
Klucze	209
Schematy	209
Relacje	210
Jak zaprojektować internetową bazę danych?	211
Określ obiekty świata realnego, których model chcesz wykonać	211
Unikaj przechowywania redundantnych danych	211
Zapisuj atomowe wartości kolumn	213
Dobierz właściwe klucze	214
Pomyśl o zapytaniach, które zadasz bazie	214
Unikaj tworzenia tabel z wieloma pustymi polami	214
Typy tabel — podsumowanie	215
Architektura internetowej bazy danych	215
Propozycje dalszych lektur	216
W następnym rozdziale	216
Rozdział 9. Tworzenie internetowej bazy danych	217
Użytkowanie monitora MySQL	218
Logowanie się do serwera MySQL	219
Tworzenie baz i rejestrowanie użytkowników	220
Definiowanie użytkowników i przywilejów	220

Wprowadzenie do systemu przywilejów MySQL	221
Zasada najmniejszego przywileju	221
Rejestrowanie użytkowników: polecenie GRANT	221
Typy i poziomy przywilejów	223
Polecenie REVOKE	224
Przykłady użycia poleceń GRANT i REVOKE	225
Rejestrowanie użytkownika łączącego się z internetu	226
Używanie odpowiedniej bazy danych	226
Tworzenie tabel bazy danych	227
Znaczenie dodatkowych atrybutów kolumn	228
Typy kolumn	229
Rzut oka na bazę danych — polecenia SHOW i DESCRIBE	231
Tworzenie indeksów	232
Identyfikatory MySQL	232
Wybór typów danych w kolumnach	233
Typy liczbowe	233
Propozycje dalszych lektur	236
W następnym rozdziale	237
Rozdział 10. Praca z bazą danych MySQL	239
Czym jest SQL?	239
Zapisywanie danych do bazy	240
Wyszukiwanie danych w bazie	242
Wyszukiwanie danych spełniających określone kryteria	243
Wyszukiwanie danych w wielu tabelach	245
Szeregowanie danych w określonym porządku	249
Grupowanie i agregowanie danych	250
Wskazanie wierszy, które mają być wyświetlone	252
Używanie podzapytań	252
Dokonywanie zmian rekordów w bazie danych	255
Zmiana struktury istniejących tabel	255
Usuwanie rekordów z bazy danych	257
Usuwanie tabel	257
Usuwanie całych baz danych	258
Propozycje dalszych lektur	258
W następnym rozdziale	258
Rozdział 11. Łączenie się z bazą MySQL za pomocą PHP	259
Jak działa internetowa baza danych?	259
Wykonywanie zapytań do bazy danych z poziomu strony WWW	262
Sprawdzenie poprawności wpisanych danych	263
Ustanawianie połączenia z bazą danych	264
Wybór właściwej bazy danych	265
Wysyłanie zapytań do bazy danych	265
Odczytywanie rezultatów zapytań	266
Zamykanie połączenia z bazą danych	267
Wstawianie nowych danych do bazy	267
Używanie instrukcji przygotowywanych	270
Używanie innych interfejsów bazodanowych PHP	272
Stosowanie ogólnego interfejsu bazodanowego: PEAR MDB2	272
Propozycje dalszych lektur	274
W następnym rozdziale	274

Rozdział 12. Administrowanie MySQL dla zaawansowanych	275
Szczegóły systemu przywilejów	275
Tabela user	276
Tabele db i host	278
Tabele tables_priv, columns_priv i procs_priv	278
Kontrola dostępu: w jaki sposób MySQL używa tabel przywilejów	279
Zmiana przywilejów: kiedy zmiany zostaną uwzględnione?	281
Ochrona bazy danych	282
MySQL z perspektywy systemu operacyjnego	282
Hasła	282
Przywileje użytkowników	283
MySQL i internet	283
Uzyskiwanie szczegółowych informacji o bazie danych	284
Uzyskiwanie informacji poleceniem SHOW	284
Uzyskiwanie informacji o kolumnach za pomocą polecenia DESCRIBE	286
Jak wykonywane są zapytania: polecenie EXPLAIN	286
Optymalizowanie bazy danych	291
Optymalizacja projektu bazy danych	291
Przywileje	291
Optymalizacja tabel	291
Stosowanie indeksów	292
Używanie wartości domyślnych	292
Więcej wskazówek	292
Tworzenie kopii zapasowej bazy danych MySQL	292
Przywracanie bazy danych MySQL	293
Implementowanie replikacji	293
Konfigurowanie serwera nadrzędnego	294
Transfer danych początkowych	294
Konfigurowanie odbiorcy lub odbiorców	295
Propozycje dalszych lektur	296
W następnym rozdziale	296
Rozdział 13. Zaawansowane programowanie w MySQL	297
Instrukcja LOAD DATA INFILE	297
Maszyny zapisu	298
Transakcje	299
Definicje dotyczące transakcji	299
Użycie transakcji w InnoDB	300
Klucze obce	301
Procedury składowane	302
Prosty przykład	302
Zmienne lokalne	304
Kursory i struktury sterujące	305
Propozycje dalszych lektur	308
W następnym rozdziale	308
Część III E-commerce i bezpieczeństwo	309
Rozdział 14. Komercyjne witryny internetowe	311
Co chcemy osiągnąć?	311
Rodzaje komercyjnych stron WWW	311
Publikowanie informacji w broszurach internetowych	312
Przyjmowanie zamówień na produkty i usługi	315
Dostarczanie usług lub wyrobów mających postać cyfrową	319

	Zwiększanie wartości produktów i usług	319
	Ograniczanie kosztów	320
	Ryzyko i zagrożenia	320
	Crackerzy	321
	Przyciągnięcie niewystarczającej liczby klientów	321
	Awarie sprzętu komputerowego	322
	Awarie sieci elektrycznych, komunikacyjnych i komputerowych oraz systemu wysyłkowego	322
	Silna konkurencja	322
	Błędy w oprogramowaniu	323
	Zmiany polityki rządowej	323
	Ograniczenie pojemności systemów	323
	Wybór strategii	323
	W następnym rozdziale	324
Rozdział 15.	Bezpieczeństwo komercyjnych stron WWW	325
	Jaką wagę mają posiadane przez nas informacje?	326
	Zagrożenia bezpieczeństwa	326
	Ujawnienie informacji poufnych	327
	Utrata lub zniszczenie danych	328
	Modyfikacje danych	329
	Blokada usługi	330
	Błędy w oprogramowaniu	331
	Zaprzeczenie korzystania z usługi	332
	Użyteczność, wydajność, koszty i bezpieczeństwo	333
	Opracowanie polityki bezpieczeństwa	333
	Zasady uwierzytelniania	334
	Podstawy szyfrowania	335
	Szyfrowanie z kluczem prywatnym	336
	Szyfrowanie z kluczem publicznym	337
	Podpis cyfrowy	338
	Certyfikaty cyfrowe	339
	Bezpieczne serwery WWW	339
	Monitorowanie i zapisywanie zdarzeń	340
	Zapory sieciowe	341
	Tworzenie kopii zapasowych	342
	Tworzenie kopii zapasowych zwykłych plików	342
	Tworzenie kopii zapasowych i odzyskiwanie baz danych MySQL	342
	Bezpieczeństwo fizyczne	343
	W następnym rozdziale	343
Rozdział 16.	Bezpieczeństwo aplikacji internetowych	345
	Strategie zapewniania bezpieczeństwa	345
	Planowanie z wyprzedzeniem	346
	Równowaga między bezpieczeństwem i użytecznością	346
	Monitorowanie bezpieczeństwa	347
	Ogólne podejście do bezpieczeństwa	347
	Rozpoznawanie zagrożeń	347
	Dostęp do danych poufnych i ich modyfikowanie	347
	Utrata lub zniszczenie danych	348
	Zablokowanie usługi	348
	Wstrzykiwanie kodu	349
	Złamanie zabezpieczeń dostępu do serwera	349
	Identyfikacja użytkowników	349
	Crackerzy	350
	Nieświadomi użytkownicy zainfekowanych komputerów	350

Rozczarowani pracownicy	350
Złodzieje sprzętu komputerowego	350
Autorzy systemów	350
Zabezpieczanie kodu źródłowego	351
Filtrowanie danych pochodzących od użytkowników	351
Unieważnianie danych wynikowych	355
Organizacja kodu źródłowego	356
Zawartość kodu źródłowego	357
Zagadnienia dotyczące systemu plików	358
Stabilność kodu i błędy	358
Apostrofy wykonywania poleceń systemu operacyjnego i polecenie exec	359
Zabezpieczanie serwera WWW oraz PHP	360
Regularne uaktualnianie oprogramowania	361
Analiza ustawień w pliku php.ini	362
Konfiguracja serwera WWW	362
Aplikacje internetowe działające na serwerach komercyjnych	364
Bezpieczeństwo serwera bazy danych	365
Użytkownicy i system uprawnień	365
Wysyłanie danych do serwera	366
Łączenie się z serwerem	366
Praca serwera	367
Zabezpieczanie sieci	367
Instalacja zapory sieciowej	368
Wykorzystanie strefy zdemilitaryzowanej	368
Przygotowanie na ataki DoS i DDoS	369
Bezpieczeństwo komputerów i systemów operacyjnych	369
Uaktualnianie systemu operacyjnego	369
Udostępnianie tylko niezbędnych usług	370
Fizyczne zabezpieczenie serwera	370
Planowanie działań na wypadek awarii	371
W następnym rozdziale	372
Rozdział 17. Uwierzytelnianie przy użyciu PHP i MySQL	373
Identyfikacja użytkowników	373
Implementacja kontroli dostępu	374
Przechowywanie haseł dostępu	376
Szyfrowanie haseł	378
Zastrzeżenie więcej niż jednej strony	379
Podstawowa metoda uwierzytelniania	380
Wykorzystanie podstawowej metody uwierzytelniania w PHP	381
Wykorzystanie podstawowej metody uwierzytelniania na serwerze Apache przy użyciu plików .htaccess	383
Wykorzystanie modułu mod_auth_mysql do celów uwierzytelniania	386
Instalacja modułu mod_auth_mysql	386
Praca z mod_auth_mysql	387
Implementacja własnej metody uwierzytelniania	388
Propozycje dalszych lektur	388
W następnym rozdziale	389
Rozdział 18. Zabezpieczanie transakcji przy użyciu PHP i MySQL	391
Zapewnienie bezpieczeństwa transakcji	391
Komputer użytkownika	392
Internet	393
System docelowy	394
Wykorzystanie protokołu Secure Sockets Layer (SSL)	395
Kontrola danych pochodzących od użytkownika	398

	Bezpieczne przechowywanie danych	399
	Ustalanie, czy powinno się przechowywać numery kart kredytowych	400
	Szyfrowanie danych w PHP	401
	Instalacja GPG	401
	Testowanie GPG	404
	Propozycje dalszych lektur	408
	W następnej części	408
Część IV	Zaawansowane techniki PHP	409
Rozdział 19.	Interakcja z systemem plików i serwerem	411
	Wprowadzenie do wysyłania plików	411
	Kod HTML służący do wysyłania plików	412
	Tworzenie PHP obsługującego plik	413
	Najczęściej spotykane problemy	417
	Stosowanie funkcji katalogowych	418
	Odczyt z katalogów	418
	Otrzymywanie informacji na temat aktualnego katalogu	421
	Tworzenie i usuwanie katalogów	421
	Interakcja z systemem plików	422
	Otrzymywanie informacji o pliku	422
	Zmiana właściwości pliku	424
	Tworzenie, usuwanie i przenoszenie plików	425
	Stosowanie funkcji uruchamiających programy	425
	Interakcja ze środowiskiem: funkcje getenv() i putenv()	427
	Propozycje dalszych lektur	428
	W następnym rozdziale	428
Rozdział 20.	Stosowanie funkcji sieci i protokołu	429
	Przegląd protokołów	429
	Wysyłanie i odczytywanie poczty elektronicznej	430
	Korzystanie z danych z innych witryn WWW	430
	Stosowanie funkcji połączeń sieciowych	433
	Tworzenie kopii bezpieczeństwa lub kopii lustrzanej pliku	436
	Stosowanie FTP w celu utworzenia kopii bezpieczeństwa lub kopii lustrzanej pliku	436
	Wysyłanie plików	442
	Unikanie przekroczenia dopuszczalnego czasu	442
	Stosowanie innych funkcji FTP	443
	Propozycje dalszych lektur	443
	W następnym rozdziale	444
Rozdział 21.	Zarządzanie datą i czasem	445
	Uzyskiwanie informacji o dacie i czasie w PHP	445
	Stosowanie funkcji date()	445
	Obsługa znaczników czasu Uniksa	447
	Stosowanie funkcji getdate()	448
	Sprawdzanie poprawności dat przy użyciu funkcji checkdate()	449
	Formatowanie znaczników czasu	450
	Konwersja pomiędzy formatami daty PHP i MySQL	450
	Obliczanie dat w PHP	452
	Obliczanie dat w MySQL	454
	Stosowanie mikrosekund	455
	Stosowanie funkcji kalendarzowych	455
	Propozycje dalszych lektur	456
	W następnym rozdziale	456

Rozdział 22.	Generowanie obrazków	457
	Konfigurowanie obsługi obrazków w PHP	457
	Formaty obrazków	458
	JPEG	459
	PNG	459
	WBMP	459
	GIF	459
	Tworzenie obrazków	460
	Tworzenie kadru obrazka	460
	Rysowanie lub umieszczanie tekstu w obrazku	461
	Wyświetlanie ostatecznej grafiki	463
	Końcowe czynności porządkujące	464
	Stosowanie automatycznie generowanych obrazków na innych stronach	464
	Stosowanie tekstu i czcionek do tworzenia obrazków	465
	Konfiguracja podstawowego kadru	467
	Dopasowanie tekstu do przycisku	468
	Nadawanie tekstowi odpowiedniej pozycji	470
	Wpisywanie tekstu do przycisku	471
	Etap końcowy	471
	Rysowanie figur i wykresów danych	471
	Inne funkcje obrazków	478
	Propozycje dalszych lektur	478
	W następnym rozdziale	478
Rozdział 23.	Stosowanie kontroli sesji w PHP	479
	Czym jest kontrola sesji?	479
	Podstawowa zasada działania sesji	479
	Czym jest cookie?	480
	Konfiguracja cookies w PHP	480
	Stosowanie cookies w sesji	481
	Przechowywanie identyfikatora sesji	481
	Implementacja prostych sesji	482
	Rozpoczynanie sesji	482
	Zgłaszanie zmiennych sesji	482
	Stosowanie zmiennych sesji	483
	Usuwanie zmiennych i niszczenie sesji	483
	Przykład prostej sesji	483
	Konfiguracja kontroli sesji	485
	Implementacja uwierzytelniania w kontroli sesji	485
	Propozycje dalszych lektur	491
	W następnym rozdziale	491
Rozdział 24.	Inne przydatne własności	493
	Stosowanie magicznych cudzysłowów	493
	Wykonywanie ciągów — funkcja eval()	494
	Zakończenie wykonania — die i exit	495
	Serializacja zmiennych i obiektów	495
	Pobieranie informacji na temat środowiska PHP	496
	Uzyskiwanie informacji na temat załadowanych rozszerzeń	496
	Identyfikacja właściciela skryptu	497
	Uzyskiwanie informacji na temat daty modyfikacji skryptu	497
	Czasowa zmiana środowiska wykonawczego	497
	Podświetlanie źródeł	498
	Używanie PHP w wierszu poleceń	499
	W następnej części	500

Część V	Tworzenie praktycznych projektów PHP i MySQL	501
Rozdział 25.	Stosowanie PHP i MySQL w dużych projektach	503
	Zastosowanie inżynierii oprogramowania w tworzeniu aplikacji WWW	504
	Planowanie i prowadzenie projektu aplikacji WWW	504
	Ponowne stosowanie kodu	505
	Tworzenie kodu łatwego w utrzymaniu	506
	Standardy kodowania	506
	Dzielenie kodu	509
	Stosowanie standardowej struktury katalogów	509
	Dokumentacja i dzielenie wewnętrznych funkcji	510
	Implementacja kontroli wersji	510
	Wybór środowiska programistycznego	511
	Dokumentacja projektów	511
	Prototypowanie	512
	Oddzielanie logiki i zawartości	513
	Optymalizacja kodu	514
	Stosowanie prostych optymalizacji	514
	Stosowanie produktów firmy Zend	514
	Testowanie	515
	Propozycje dalszych lektur	516
	W następnym rozdziale	516
Rozdział 26.	Usuwanie błędów	517
	Błędy programistyczne	517
	Błędy składni	517
	Błędy wykonania	519
	Błędy logiczne	523
	Pomoc w usuwaniu błędów w zmiennych	525
	Poziomy zgłaszania błędów	527
	Zmiana ustawień zgłaszania błędów	528
	Wyzwalanie własnych błędów	529
	Elegancka obsługa błędów	529
	W następnym rozdziale	532
Rozdział 27.	Tworzenie uwierzytelniania użytkowników i personalizacji	533
	Składniki rozwiązania	533
	Identyfikacja użytkownika i personalizacja	534
	Przechowywanie zakładek	535
	Rekomendowanie zakładek	535
	Przegląd rozwiązania	535
	Implementacja bazy danych	537
	Implementacja podstawowej witryny	538
	Implementacja uwierzytelniania użytkowników	540
	Rejestracja użytkowników	540
	Logowanie	545
	Wylogowanie	548
	Zmiana hasła	549
	Ustawianie zapomnianych haseł	551
	Implementacja przechowywania i odczytywania zakładek	555
	Dodawanie zakładek	555
	Wyświetlanie zakładek	557
	Usuwanie zakładek	557
	Implementacja rekomendacji	559
	Rozwijanie projektu i możliwe rozszerzenia	562
	W następnym rozdziale	562

Rozdział 28. Tworzenie koszyka na zakupy	563
Składniki rozwiązania	563
Tworzenie katalogu online	564
Śledzenie zakupów użytkownika podczas przeglądania	564
Implementacja systemu płatności	564
Interfejs administratora	565
Przegląd rozwiązania	565
Implementacja bazy danych	568
Implementacja katalogu online	570
Przedstawianie kategorii	571
Wyświetlanie książek danej kategorii	574
Przedstawianie szczegółowych danych książki	575
Implementacja koszyka na zakupy	577
Stosowanie skryptu pokaz_kosz.php	577
Podgląd koszyka	580
Dodawanie produktów do koszyka	582
Zapisywanie uaktualnionego koszyka	583
Wyświetlanie podsumowania w pasku nagłówka	584
Pobyt w kasie	584
Implementacja płatności	589
Implementacja interfejsu administratora	591
Rozwijanie projektu	598
Zastosowanie istniejącego systemu	598
W następnym rozdziale	598
 Rozdział 29. Tworzenie serwisu poczty elektronicznej opartego na WWW	 599
Składniki rozwiązania	599
Protokoły poczty: POP3 i IMAP	599
Obsługa POP3 i IMAP w PHP	600
Przegląd rozwiązania	601
Konfiguracja bazy danych	603
Architektura skryptu	604
Logowanie i wylogowanie	608
Konfiguracja kont	611
Tworzenie nowego konta	613
Modyfikacja istniejącego konta	614
Usuwanie konta	614
Odczytywanie poczty	615
Wybór konta	615
Przeglądanie zawartości skrzynki	617
Odczytywanie wiadomości pocztowych	619
Przeglądanie nagłówków wiadomości	622
Usuwanie wiadomości	623
Wysyłanie wiadomości	623
Wysyłanie nowej wiadomości	624
Odpowiadanie i przekazywanie poczty	625
Rozwijanie projektu	626
W następnym rozdziale	627
 Rozdział 30. Tworzenie menedżera list pocztowych	 629
Składniki rozwiązania	629
Konfiguracja bazy danych list i abonentów	630
Wysyłanie plików	630
Wysyłanie wiadomości z załącznikami	631
Przegląd rozwiązania	631
Konfiguracja bazy danych	633

Architektura skryptu	635
Implementacja logowania	641
Tworzenie nowego konta	641
Logowanie	643
Implementacja funkcji użytkownika	645
Przeglądanie list	646
Przeglądanie informacji na temat listy	650
Przeglądanie archiwum listy	652
Zapisywanie i wypisywanie	653
Zmiana konfiguracji konta	654
Zmiana hasła	654
Wylogowanie	656
Implementacja funkcji administratora	656
Tworzenie nowej listy	657
Wysyłanie nowych wiadomości	658
Obsługa wysyłania wielu plików	661
Podgląd wiadomości	664
Rozsyłanie wiadomości	665
Rozwijanie projektu	670
W następnym rozdziale	670
Rozdział 31. Tworzenie forum WWW	671
Proces	671
Składniki rozwiązania	672
Przegląd rozwiązania	673
Projektowanie bazy danych	674
Przeglądanie drzewa artykułów	676
Rozwijanie i zwijanie	678
Wyświetlanie artykułów	680
Korzystanie z klasy <code>wezel_drzewa</code>	681
Przeglądanie pojedynczych artykułów	687
Dodawanie nowych artykułów	688
Rozszerzenia	694
Wykorzystanie istniejącego systemu	694
W następnym rozdziale	694
Rozdział 32. Tworzenie dokumentów spersonalizowanych w formacie PDF	695
Opis projektu	695
Ocena formatów dokumentów	696
Składniki rozwiązania	700
System pytań i odpowiedzi	700
Oprogramowanie generujące dokumenty	700
Przegląd rozwiązania	703
Zadawanie pytań	704
Ocena odpowiedzi	705
Tworzenie certyfikatu RTF	707
Tworzenie certyfikatu PDF z szablonu	710
Generowanie dokumentu PDF za pomocą PDFlib	713
Skrypt „Witaj, świecie” dla PDFlib	713
Tworzenie certyfikatu za pomocą PDFlib	717
Problemy związane z nagłówkami	723
Rozwijanie projektu	724
W następnym rozdziale	724

Rozdział 33.	Korzystanie z usług sieciowych za pomocą XML i SOAP	725
	Opis projektu: korzystanie z języka XML i usług sieciowych	725
	Podstawy XML	726
	Podstawy usług sieciowych	729
	Składniki rozwiązania	730
	Korzystanie z interfejsu usług sieciowych Amazon.com	730
	Wczytywanie dokumentów XML: odpowiedzi REST	731
	Korzystanie z SOAP za pomocą PHP	732
	Buforowanie	732
	Opis rozwiązania	732
	Aplikacja główna	734
	Wyświetlanie listy książek z danej kategorii	742
	Tworzenie obiektu klasy WynikiWyszukiwania	743
	Korzystanie z REST do wykonywania żądań i odczytywania wyników	750
	Korzystanie z protokołu SOAP do wykonywania żądania i odczytywania wyniku	755
	Buforowanie danych pochodzących z żądania	756
	Konstrukcja koszyka na zakupy	758
	Przejsięcie do kasy na witrynie Amazon.com	761
	Instalacja kodu źródłowego	762
	Kierunki rozwoju	762
	Literatura	762
Rozdział 34.	Tworzenie aplikacji Web 2.0 z wykorzystaniem technologii Ajax	763
	Czym jest technologia Ajax?	764
	Żądania i odpowiedzi HTTP	764
	DHTML i XML	765
	Kaskadowe arkusze stylów (CSS)	766
	Skrypty działające po stronie klienta	767
	Skrypty działające po stronie serwera	768
	XML i XSLT	768
	Podstawy technologii Ajax	768
	Obiekt XMLHttpRequest	768
	Komunikowanie się z serwerem	770
	Przetwarzanie odpowiedzi serwera	772
	Połączenie wszystkich elementów aplikacji	773
	Dodanie nowych elementów do wcześniejszych projektów	775
	Dodanie elementów Ajaksa do witryny ZakładkaPHP	777
	Źródła dodatkowych informacji	788
	Dodatkowe informacje na temat Document Object Model (DOM)	789
	Biblioteki JavaScript dla aplikacji Ajax	789
	Witryny internetowe przeznaczone dla programistów Ajax	790
Dodatki	791	
Dodatek A	Instalacja PHP i MySQL	793
	Instalacja Apache, PHP i MySQL w systemie UNIX	794
	Instalacja przy użyciu binariów	794
	Instalacja przy użyciu kodów źródłowych	794
	Plik httpd.conf — informacje końcowe	800
	Czy obsługa PHP działa poprawnie?	800
	Czy SSL działa poprawnie?	801

Instalacja Apache, PHP i MySQL w systemie Windows	802
Instalacja MySQL w systemie Windows	803
Instalacja serwera Apache w systemie Windows	804
Instalacja PHP w systemie Windows	806
Instalowanie PEAR	808
Inne konfiguracje	809
Dodatek B	
Zasoby internetowe	811
Zasoby poświęcone PHP	811
Zasoby poświęcone MySQL i SQL	813
Zasoby poświęcone serwerowi Apache	813
Zasoby poświęcone tworzeniu stron WWW	814
Skorowidz	815

Rozdział 2.

Przechowywanie i wyszukiwanie danych

W poprzednim rozdziale omówiliśmy sposoby dostępu do danych umieszczonych w formularzu HTML i metody manipulowania nimi. Teraz przedstawiamy metody zapisywania informacji w celu późniejszego ich wykorzystania. W większości przypadków, włączając w to przykład z poprzedniego rozdziału, celem jest przechowanie danych i późniejsze ich załadowanie. W tym przykładzie należy zapamiętać zamówienie klienta, aby później je zrealizować.

W tym rozdziale opiszemy sposoby zapisania do pliku zamówienia przedstawionego w przykładzie oraz metody późniejszego odczytania tego pliku. Pokażemy również, dlaczego takie rozwiązanie nie zawsze jest najlepsze. Pracując z większą liczbą zamówień, powinno się zamiast niego używać systemu zarządzania bazami danych, takiego jak MySQL.

W tym rozdziale zostaną poruszone następujące zagadnienia:

- zapisywanie danych do późniejszego użycia,
- otwieranie pliku,
- tworzenie i zapisywanie pliku,
- zamykanie pliku,
- czytanie z pliku,
- blokowanie pliku,
- usuwanie pliku,
- inne przydatne informacje na temat plików,
- lepszy sposób obróbki danych: systemy zarządzania bazami danych.

Zapisywanie danych do późniejszego użycia

Istnieją dwa sposoby przechowywania danych — w pliku jednorodnym oraz w bazie danych.

Plik jednorodny może mieć wiele różnych formatów, lecz zazwyczaj terminem tym oznacza się prosty plik tekstowy. W opisywanym przykładzie dane są zapisywane w pliku tekstowym, jedno zamówienie w jednym wierszu.

Zapisywanie zamówień w taki właśnie sposób jest rozwiązaniem bardzo prostym w realizacji, ale zarazem jest ono obciążone licznymi ograniczeniami, co zostanie pokazane w dalszej części rozdziału. Przy obróbce danych znacznej wielkości stosuje się zazwyczaj bazy danych. Mimo to pliki jednorodne znajdują zastosowania i istnieją przypadki, w których wiedza na ich temat jest konieczna.

Proces zapisu i odczytu plików w PHP przebiega w zbliżony sposób jak w wielu innych, podobnych językach programowania. Osoby znające język C lub skrypty powłoki Uniksa powinny bez trudu rozpoznać podobieństwa tych procedur.

Przechowywanie i wyszukiwanie zamówień Janka

Poniżej użyta zostanie nieco zmodyfikowana wersja formularza zamówień, przedstawionego w poprzednim rozdziale. Na początku należy przeanalizować ten formularz i kod PHP stworzony w celu obróbki zamówień.



Kod HTML i skrypty PHP zastosowane w tym rozdziale znajdują się w folderze *rozdzial_02* (przykłady są dostępne na płycie CD dołączonej do książki).

Formularz został zmodyfikowany w celu łatwego uzyskania adresu klienta. Nowa wersja formularza jest przedstawiona na rysunku 2.1.

Rysunek 2.1.
Wersja formularza zamówień pobierająca również adres klienta

The screenshot shows a web browser window titled "Części samochodowe Janka - Windows Internet Explorer". The address bar shows "http://localhost/rozdzial_02/formularz.html". The page content includes a title "Części samochodowe Janka" and a subtitle "Formularz zamówienia". Below this is a form with the following fields:

Produkt	Ilość
Opony	<input type="text"/>
Olej	<input type="text"/>
Świece Zapłonowe	<input type="text"/>
Adres	<input type="text"/>

At the bottom of the form is a button labeled "Złóż zamówienie". The browser's status bar at the bottom indicates "Lokalny intranet" and "100%".

Pole formularza zawierające adres klienta nosi nazwę `adres`. Podczas przetwarzania w PHP daje ono zmienną o nazwie `$adres`, pod warunkiem stosowania stylu krótkiego dostępu do zmiennych. Należy pamiętać, że przy zastosowaniu stylu długiego odwołanie do tej zmiennej to `$REQUEST['adres']`, `$_POST['adres']` lub `$_GET['adres']`, zależnie od metody zatwierdzenia (szczegóły opisaliśmy w rozdziale 1.).

W tym rozdziale każde nadchodzące zamówienie zostanie zapisane w tym samym pliku. Skonstruowany później interfejs WWW pozwoli pracownikom Janka na przeglądanie przyjętych zamówień.

Przetwarzanie plików

Zapisywanie danych w pliku następuje w trzech etapach. Są to:

1. Otwarcie pliku. Jeżeli dany plik nie istnieje, należy go utworzyć.
2. Zapisanie danych w pliku.
3. Zamknięcie pliku.

Podobnie, trój etapowo, przebiega odczytywanie danych z pliku:

1. Otwarcie pliku. Jeżeli plik nie może zostać otwarty (np. nie istnieje), fakt ten musi zostać rozpoznany i program powinien zakończyć się w elegancki sposób (tzn. nie bombardując użytkownika dokładnymi i niepotrzebnymi mu informacjami o błędach).
2. Odczytanie danych z pliku.
3. Zamknięcie pliku.

Przy odczytywaniu danych z pliku dostępnych jest wiele sposobów ustalania ilości pobieranych naraz danych. Rozwiązania najczęściej stosowane zostaną opisane bardziej szczegółowo w poniższych punktach. Na początek przedstawimy mechanizm otwierania plików.

Otwieranie pliku

Aby otworzyć plik w PHP, stosuje się funkcję `fopen()`. Otwierając plik, należy zadeklarować sposób, w jaki będzie on używany. Sposób ten nosi nazwę *trybu otwarcia pliku*.

Tryby otwarcia pliku

System operacyjny serwera musi mieć informacje na temat przeznaczenia otwieranego pliku. Musi wiedzieć, czy plik może równocześnie zostać otwarty przez inny skrypt oraz czy użytkownik posiada uprawnienia do dostępu i modyfikacji pliku. Przede wszystkim tryb otwarcia pliku dostarcza systemowi operacyjnemu mechanizmu przetwarzania żądań dostępu od innych użytkowników bądź skryptów oraz metody sprawdzania uprawnień dostępu do konkretnych plików.

Przy otwieraniu pliku należy mieć trzy informacje:

1. Można otworzyć plik w następujących trybach: tylko do odczytu, tylko do zapisu lub do obu tych celów.
2. Przy zapisywaniu danych w pliku można nadpisać istniejące dane bądź dodać nowe na jego końcu. Można również opracować zgrabny sposób zakańczania programu zamiast nadpisywania pliku na pliku, który już istnieje.
3. Przy zapisywaniu pliku przy użyciu systemu rozróżniającego pliki tekstowe i binarne można określić dany typ.

Funkcja `fopen()` rozpoznaje połączenia tych trzech opcji.

Stosowanie funkcji fopen() do otwarcia pliku

Aby zapisać zamówienie klienta do pliku zamówień Janka, należy zastosować następujący wiersz kodu:

```
$wp = fopen("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt", 'w');
```

Przy wywołaniu funkcja `fopen` spodziewa się dwóch lub trzech parametrów. Zazwyczaj stosuje się dwa, jak pokazano w powyższym przykładzie.

Pierwszy parametr to nazwa pliku, który ma zostać otwarty. Można tu określić ścieżkę dostępu do pliku, jak w powyższym przykładzie; plik *zamowienia.txt* znajduje się w katalogu zamówień. Zastosowana została wbudowana w PHP zmienna `$SERVER['DOCUMENT_ROOT']` lecz, ze względu na uciążliwość stosowania pełnych nazw zmiennych formy, przypisaliśmy jej krótszą nazwę.

Zmienna ta wskazuje na podstawowy element drzewa katalogów serwera WWW. W wierszu tym użyto symbolu `..`, oznaczającego „katalog nadrzędny katalogu macierzystego”, który ze względu na bezpieczeństwo znajduje się poza drzewem katalogów. Nie można pozwolić na inny sposób dostępu przez WWW do tego pliku poza dostarczonym interfejsem. Ścieżka tego typu jest nazywana *ścieżką względną*, ponieważ opisuje miejsce w systemie plików w zależności od katalogu macierzystego.

Podobnie jak w przypadku nadawania zmiennym formy krótkich nazw, na początku skryptu należy umieścić następujący wiersz:

```
$DOCUMENT_ROOT = $_SERVER['DOCUMENT_ROOT'];
```

w celu skopiowania zawartości zmiennej noszącej nazwę długą do zmiennej o krótkiej nazwie.

Analogicznie do różnorodności metod dostępu do danych formy, istnieją również różne sposoby dostępu do predefiniowanych zmiennych serwera. W zależności od ustawień serwera (szczegółowe informacje na ten temat można znaleźć w rozdziale 1.) można pozyskać katalog macierzysty za pomocą:

- `$_SERVER['DOCUMENT_ROOT']`
- `$DOCUMENT_ROOT`
- `$HTTP_SERVER_VARS['DOCUMENT_ROOT']`

Podobnie jak w przypadku danych formy, zalecany jest pierwszy sposób.

Można również określić bezwzględną ścieżkę dostępu do pliku, będącą ścieżką od katalogu głównego (`/` w systemach Unix i zazwyczaj `C:\` w systemach Windows). Na przykładowym serwerze Uniksa ścieżka ta może wyglądać następująco: `/home/ksiazka/zamowienia`. Niedogodnością tej metody, zwłaszcza w wypadku korzystania z obcego serwera, jest możliwość modyfikacji ścieżki bezwzględnej, co może oznaczać poważne zmiany w wielu skryptach.

Jeżeli ścieżka nie zostanie podana, PHP będzie szukał pliku i ewentualnie utworzy go w tym samym katalogu, w którym znajduje się skrypt. Może się to różnić w zależności od faktu, czy PHP jest uruchamiany poprzez jakiś skrypt CGI, i zależy od konfiguracji serwera.

W środowisku Uniksa stosuje się ukośniki (`/`), natomiast w środowisku Windows można używać lewych (`\`) lub prawych ukośników (`/`), które muszą jednak zostać oznaczone jako znaki specjalne, aby funkcja `fopen` właściwie je zinterpretowała. W tym celu należy po prostu dodać przed każdym symbolem jeszcze jeden lewy ukośnik, jak pokazano w poniższym przykładzie:

```
$wp = fopen("..\..\zamowienia\\zamowienia.txt", 'w');
```

Niewiele osób stosuje w ścieżkach dostępu w kodzie PHP znaki odwrotnych ukośników, ponieważ oznaczałoby to, że kod ten będzie działał tylko w systemach Windows. Stosowanie zwykłych ukośników pozwala na przenoszenie kodu między maszynami pracującymi w systemach Unix i Windows bez konieczności wprowadzania w nim zmian.

Drugim parametrem funkcji `fopen()` jest tryb otwarcia pliku, określający jego przeznaczenie. Powinien on zostać podany jako ciąg. W powyższym przykładzie funkcji `fopen()` zostaje przekazana wartość `w`, co oznacza otwarcie pliku do zapisu. Podsumowanie trybów otwarcia pliku przedstawiono w tabeli 2.1.

Tabela 2.1. Podsumowanie trybów otwarcia pliku w funkcji `fopen`

Tryb	Nazwa trybu	Znaczenie
r	Odczyt	Otwarcie pliku do odczytu, poczynając od początku pliku
r+	Odczyt	Otwarcie pliku do odczytu i zapisu, poczynając od początku pliku
w	Zapis	Otwarcie pliku do zapisu, poczynając od początku pliku. Jeżeli plik istnieje, bieżąca zawartość zostanie skasowana. W przeciwnym wypadku nastąpi próba jego utworzenia
w+	Zapis	Otwarcie pliku do zapisu i odczytu, poczynając od początku pliku. Jeżeli plik istnieje, bieżąca zawartość zostanie skasowana, jeżeli zaś nie, nastąpi próba jego utworzenia
x	Ostrożny zapis	Otwarcie pliku do zapisu rozpoczynającego się na początku pliku. Jeśli plik już istnieje, nie zostanie otwarty, funkcja <code>fopen()</code> zwróci wartość <code>false</code> , a PHP wygeneruje ostrzeżenie
x+	Ostrożny zapis	Otwarcie pliku do zapisu i odczytu rozpoczynającego się na początku pliku. Jeśli plik już istnieje, nie zostanie otwarty, funkcja <code>fopen()</code> zwróci wartość <code>false</code> , a PHP wygeneruje ostrzeżenie
a	Dodawanie	Otwarcie pliku do dodawania zawartości, począwszy od końca istniejącej zawartości. Jeżeli plik nie istnieje, nastąpi próba jego utworzenia
a+	Dodawanie	Otwarcie pliku do dodawania zawartości i odczytu, począwszy od końca istniejącej zawartości. Jeżeli plik nie istnieje, nastąpi próba jego utworzenia
b	Tryb binarny	Stosowany w połączeniu z jednym z powyższych typów w wypadku korzystania z systemu rozróżniającego pliki tekstowe i binarne. Windows go rozróżnia, Unix — nie. Programiści PHP zalecają, by zawsze używać tej opcji w celu zapewnienia sobie maksymalnej przenośności. Jest to tryb domyślny
t	Tryb tekstowy	Stosowany w połączeniu z jednym z powyższych trybów. Tryb ten jest dostępny jedynie w systemie Windows. Nie jest on zalecany, chyba że przed przeniesieniem kodu zostanie zamieniony na tryb <code>b</code>

Tryb otwarcia pliku zastosowany w przykładzie zależy od sposobu, w jaki system zostanie użyty. Powyżej występuje tryb `'w'`, co oznacza, że w pliku będzie mogło być zapamiętane tylko jedno zamówienie. Każde nowo przyjęte zamówienie nadpisze poprzednie. Nie jest to rozwiązanie zbyt rozsądne, więc lepiej użyć trybu dodawania (oraz, zgodnie z zaleceniem, trybu binarnego):

```
$wp = fopen("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt", 'ab');
```

Istnieje również trzeci, opcjonalny parametr funkcji `fopen()`. Stosuje się go w celu szukania pliku w lokalizacjach podanych w opcji `include_path` (ustawianej w konfiguracji PHP — szczegóły w dodatku A). Aby użyć tej opcji, należy nadać temu parametrowi wartość `1`. Nie trzeba wtedy podawać ścieżki dostępu do pliku.

```
$wp = fopen('zamowienia.txt', 'ab', true);
```

Czwarty parametr również jest opcjonalny. Funkcja `fopen()` dopuszcza, by nazwy plików były poprzedzone nazwą protokołu (na przykład `http://`), a same pliki były otwierane ze zdalnych

lokalizacji. Niektóre protokoły pozwalają ponadto na przekazywanie dodatkowych parametrów. Taki sposób użycia funkcji `fopen()` zostanie opisany bardziej szczegółowo w dalszej części tego rozdziału.

Jeżeli funkcji `fopen()` uda się otwarcie pliku, zwraca ona zasób będący w rzeczywistości uchwytem albo wskaźnikiem pliku i przechowuje go w zmiennej, w powyższym przykładzie: `$wp`. Zmienna ta jest stosowana przy kolejnych próbach dostępu do pliku, to znaczy przy odczytywaniu lub zapisywaniu danych.

Otwieranie pliku przez protokół FTP lub HTTP

Funkcja `fopen()` służy do otwierania do odczytu lub zapisu plików lokalnych. Za jej pomocą można także otwierać pliki poprzez FTP, HTTP i inne protokoły. Własność tę można zablokować, wyłączając w pliku `php.ini` dyrektywę `allow_url_fopen`. Jeżeli więc otwieranie plików zdalnych przy użyciu `fopen()` sprawia kłopoty, najpierw należy zajrzeć do pliku `php.ini`.

Jeżeli wprowadzona nazwa pliku rozpoczyna się od `ftp://`, otwarte zostanie pasywne połączenie FTP z serwerem, którego adres został wprowadzony, a funkcja zwróci wartość wskaźnika na początek pliku.

Jeżeli wprowadzona nazwa pliku rozpoczyna się od `http://`, otwarte zostanie pasywne połączenie HTTP z serwerem, którego adres został wprowadzony, a funkcja zwróci wartość wskaźnika na odpowiedź. Przy zastosowaniu trybu HTTP w starszych wersjach PHP adres odnoszący się do katalogu musi zawierać kończące ukośniki, jak w poniższym przykładzie:

```
http://www.serwer.com/
```

a nie

```
http://www.serwer.com
```

Przy zastosowaniu drugiej wersji adresu (bez ukośnika) serwer WWW użyje zwykłego przekierowania HTTP i prześle w odpowiedzi pierwszy z powyższych adresów (warto wykonać to zadanie).

Należy pamiętać, że nazwy domen w URL-ach nie są różnicowane ze względu na wielkość liter, w przeciwieństwie do ścieżek i nazw plików.

Problemy z otwieraniem plików

Popularnym błędem jest próba otwarcia pliku, co do którego nie posiada się praw odczytu lub zapisu. (Błąd taki pojawia się zazwyczaj w systemach operacyjnych z rodziny Unix, od czasu do czasu można jednak spotkać się z nim w systemie Windows.) W takim przypadku PHP wyświetli ostrzeżenie podobne do przedstawionego na rysunku 2.2.

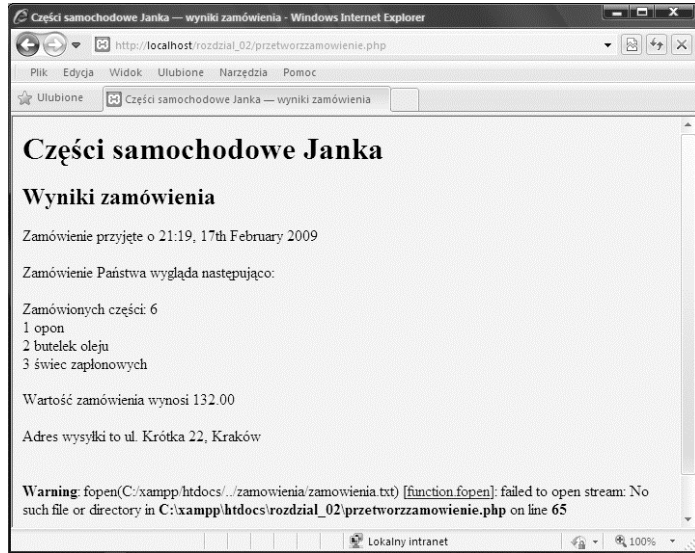
Po popełnieniu takiego błędu należy upewnić się, czy skrypt, który jest stosowany, posiada prawo dostępu do danego pliku. Zależnie od konfiguracji serwera, skrypt może być uruchomiony z prawami użytkownika serwera WWW lub z prawami właściciela swojego katalogu.

W większości systemów skrypt zostanie uruchomiony jako użytkownik serwera WWW. Jeżeli na przykład skrypt znajduje się w systemie uniksowym w katalogu `~/public_html/rozdzial2`, należy utworzyć ogólnodostępny katalog, w którym przechowywane będą zamówienia. Aby to uczynić, można wpisać:

```
mkdir ~/zamowienia
chmod 777 ~/zamowienia/
```


Rysunek 2.2.

Podczas nieudanej próby otwarcia pliku PHP wyświetla specyficzne ostrzeżenie



Należy pamiętać, że katalogi i pliki z ogólnym prawem zapisu są bardzo niebezpieczne. W szczególności nie powinno się używać katalogów dostępnych bezpośrednio z poziomu WWW, które posiadają możliwość zapisu. Z tego powodu przykładowy katalog *zamowienia* został umieszczony dwa poziomy wyżej, ponad katalogiem *public_html*. Szczegółowe informacje na temat bezpieczeństwa są przedstawione w rozdziale 15.

Złe ustawienia dostępu do plików to najpopularniejszy, lecz nie jedyny błąd popełniany przy otwieraniu plików. Jeżeli plik nie może zostać otwarty, trzeba koniecznie o tym wiedzieć, aby nie próbować odczytywać ani zapisywać w nim danych.

Jeżeli wywołanie funkcji `fopen()` nie powiedzie się, zwróci ona wartość `false`. Można wtedy zastąpić oryginalny komunikat o błędzie PHP innym, bardziej przyjaznym dla użytkownika:

```
@ $wp = fopen("$DOCUMENT_ROOT/../../zamowienia/zamowienia.txt", 'ab');

if (!$wp) {
    echo "<p><strong> Zamówienie Państwa nie może zostać przyjęte w tej chwili.
        Proszę spróbować później.</strong></p></body></html>";
    exit;
}
```

Symbol `@` umieszczony przed wywołaniem funkcji `fopen()` nakazuje PHP wytłumienie wszystkich błędów wynikłych z tego wywołania. Zazwyczaj warto wiedzieć, kiedy występuje błąd. Kwestię tę rozważymy później.

Wiersz ten można również zapisać w następujący sposób:

```
$wp = @fopen("$DOCUMENT_ROOT/../../ zamowienia/zamowienia.txt". 'a');
```

jednak w takiej sytuacji nie widać wyraźnie, że stosowane jest ukrywanie błędów, co może utrudnić debugowanie kodu.

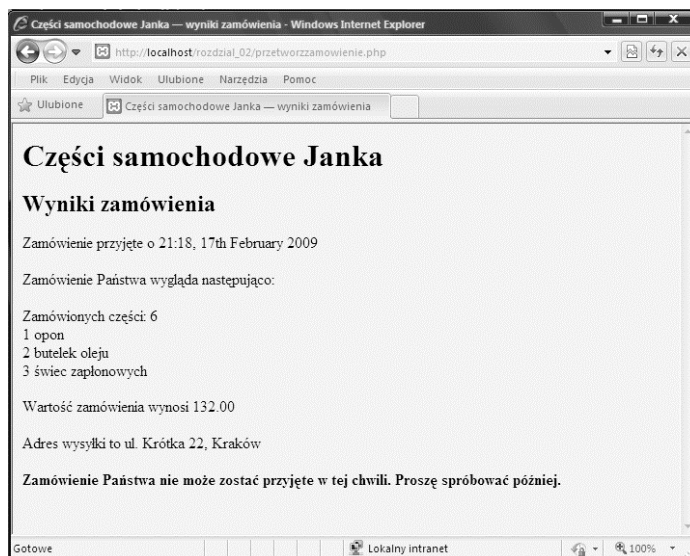
Opisana metoda stanowi najprostszy sposób radzenia sobie z błędami. Bardziej elegancki sposób obsługi błędów zostanie przedstawiony w rozdziale 7.

Instrukcja `if` sprawdza wartość zmiennej `$wp`, aby ustalić, czy wywołanie funkcji `fopen()` zwróciło prawidłowy wskaźnik. Jeżeli nie, wyświetla komunikat o błędzie i kończy działanie skryptu. Ponieważ strona zakończy się w tym miejscu, w powyższym kodzie zamknięte zostały również znaczniki HTML, aby kod HTML działał bezbłędnie.

Wynik działania powyższego fragmentu skryptu został przedstawiony na rysunku 2.3.

Rysunek 2.3.

Stosowanie własnych komunikatów o błędach zamiast tych wbudowanych w PHP jest niewątpliwie bardziej przyjazne dla użytkownika



Zapisywanie danych w pliku

Zapisywanie danych w pliku w PHP jest stosunkowo proste. Stosuje się do tego funkcję `fwrite()` (zapis do pliku) lub `fputs()` (umieszczenie ciągu w pliku). Funkcja `fputs()` jest inną nazwą funkcji `fwrite()`. Funkcję `fwrite()` wywołuje się w następujący sposób:

```
fwrite($wp, $ciagwyjsciowy);
```

Funkcja ta nakazuje PHP zapisanie ciągu zawartego w zmiennej `$ciagwyjsciowy` do pliku wskazywanego przez zmienną `$wp`.

Alternatywą dla funkcji `fwrite()` jest funkcja `file_put_contents()`. Jej prototyp przedstawia się następująco:

```
int file_put_contents ( string nazwa_pliku,
                      string dane
                      [, int znaczniki
                      [, resource kontekst]])
```

Funkcja zapisuje ciąg znaków zawarty w danych do pliku o nazwie `nazwa_pliku`, bez potrzeby wywoływania funkcji `fopen()` (ani `fclose()`). Jest to nowa funkcja, wprowadzona w PHP5, a funkcją dla niej komplementarną jest `file_get_contents()`, o której niedługo powiemy. Opcjonalnych parametrów `znaczniki` i `kontekst` najczęściej używa się w trakcie zapisywania do plików zdalnych przy użyciu na przykład HTTP i FTP. (Funkcje te zostaną omówione w rozdziale 20.)

Parametry funkcji fwrite()

Funkcja fwrite() pobiera trzy parametry, lecz ostatni z nich jest opcjonalny. Oto prototyp funkcji fwrite():

```
int fwrite(resource wskaznik_pliku, string ciag, int [dlugosc]);
```

Trzeci parametr, *dlugosc*, zawiera maksymalną możliwą do zapisania liczbę bajtów. Jeżeli parametr ten został umieszczony w wywołaniu funkcji, fwrite() będzie zapisywać *ciag* w pliku wskazanym przez *wskaznik pliku*, dopóki nie osiągnie końca *ciagu* lub zapisze *dlugosc* bajtów, zależnie od tego, co wystąpi wcześniej.

Długość łańcucha znaków można odczytać, używając funkcji PHP o nazwie strlen() w następujący sposób:

```
fwrite($wp, $ciagwyjsciovy, strlen($ciagwyjsciovy));
```

Trzeciego parametru używa się w trakcie zapisywania w trybie binarnym, ponieważ można dzięki niemu uniknąć pewnych komplikacji związanych ze zgodnością między platformami.

Formaty plików

Tworząc plik danych podobny do przykładowego, można określić dowolny format przechowywania danych. Jeżeli jednak dane te będą wykorzystywane później przez jakąś aplikację, należy zastosować się do zasad określonych przez tę aplikację.

Poniżej przedstawiono ciąg opisujący jeden rekord w pliku danych:

```
$ciagwyjsciovy = $data."\\t".$iloscopon." opo\\t".$iloscroleju." butelek oleju\\t"
                ".$iloscswiec." swiec zaplonowych\\t".$wartosc
                ".$PLN\\t".$adres."\\n";
```

W tym prostym przykładzie każdy rekord jest zapisany w osobnym wierszu pliku. Metodę tę zastosowano, ponieważ występuje w niej prosty separator rekordów: znak nowego wiersza. Znaki te przedstawia się za pomocą sekwencji "\\n", gdyż są niewidzialne.

W całej książce pola danych będą zapisywane za każdym razem w jednakowym porządku i oddzielane znakami tabulacji. Ponieważ znak ten również jest niewidzialny, przedstawia się go za pomocą sekwencji "\\t". Można wybrać dowolny, czytelny znak podziału.

Znak podziału powinien być znakiem, który nie występuje pośród wprowadzanych danych, lub też dane powinny zostać przekształcone w celu usunięcia występujących w nich znaków podziału. Przekształcanie danych zostanie omówione w rozdziale 4. Na razie należy przyjąć, że przy wprowadzaniu zamówienia nie zostanie użyty znak tabulacji, co jest zdarzeniem możliwym, lecz mało prawdopodobnym.

Stosowanie specjalnych znaków separujących pola pozwala na łatwiejsze rozdzielenie zmiennych przy odczytywaniu danych. Kwestia ta zostanie rozważona w rozdziale 3. oraz w rozdziale 4. Tymczasem każde zamówienie będzie traktowane jako pojedynczy ciąg.

Po przyjęciu kilku zamówień zawartość pliku powinna wyglądać podobnie do przedstawionej na listingu 2.1.

Listing 2.1. *zamowienia.txt* — przykład pliku zamówień

```

19:35, 18 lipca 2008 4 opon 1 butelek oleju 6 świec zapłonowych 1820.00PLN
↳ul. Krótka 22, Kraków
19:37, 18 lipca 2008 1 opon 0 butelek oleju 0 świec zapłonowych 400.00PLN
↳ul. Główna 33, Gliwice
19:38, 18 lipca 2008 0 opon 1 butelek oleju 4 świec zapłonowych 180.00PLN
↳ul. Akacjowa 127, Warszawa

```

Zamykanie pliku

Po zakończeniu korzystania z pliku należy go zamknąć. Stosuje się w tym celu funkcję `fclose()` w następujący sposób:

```
fclose($wp);
```

Funkcja ta zwraca wartość `true`, jeżeli zamykanie powiodło się, lub `false`, w przeciwnym wypadku. Działanie to ma znacznie większe szanse powodzenia niż otwieranie pliku — i w tym przypadku nie zdecydowano się na jego sprawdzenie.

Pełen listing ostatecznej wersji skryptu *przetworzzamowienie.php* przedstawiono na listingu 2.2.

Listing 2.2. *przetworzzamowienie.php* — Ostateczna wersja skryptu przetwarzającego zamówienia

```

<?php
// utworzenie krótkich nazw zmiennych
$iloscocon = $_POST['iloscocon'];
$iloscoleju = $_POST['iloscoleju'];
$iloscswiec = $_POST['iloscswiec'];
$adres = $_POST['adres'];
$DOCUMENT_ROOT = $_SERVER['DOCUMENT_ROOT'];
$data=date('H:i, jS F Y');
?>

<html>
<head>
<title>Części samochodowe Janka – wyniki zamówienia</title>
</head>
<body>
<h1>Części samochodowe Janka</h1>
<h2>Wyniki zamówienia</h2>
<?php

echo "<p>Zamówienie przyjęte o ".$data."</p>";

echo "<p>Zamówienie Państwa wygląda następująco: </p>";

$ilosc = 0;
$ilosc = $iloscocon + $iloscoleju + $iloscswiec;
echo "Zamówionych części: ".$ilosc."<br />";

if($ilosc == 0) {
    echo "Na poprzedniej stronie nie zostało złożone żadne zamówienie!<br />";
} else {

    if ($iloscocon > 0) {
        echo $iloscocon." opon<br />";

```

```
}

if ($iloscoleju > 0) {
    echo $iloscoleju." butelek oleju<br />";
}

if ($iloscswiec > 0) {
    echo $iloscswiec." świec zapłonowych<br />";
}
}

$wartosc=0.00;

define('CENAOPON', 100);
define('CENAOLEJU', 10);
define('CENASWIEC', 4);

$wartosc =$iloscopon * CENAOPON + $iloscoleju * CENAOLEJU + $iloscswiec * CENASWIEC;

$wartosc=number_format($wartosc, 2, '.', ' ');

echo "<p>Wartość zamówienia wynosi ".$wartosc."</p>";
echo "<p>Adres wysyłki to ".$adres."</p>";

$ciagwyjsciowy = $data."\t".$iloscopon." opon \t".$iloscoleju." butelek oleju\t"
                . $iloscswiec." świec zapłonowych\t".$wartosc
                ."PLN\t". $adres."\n";

// otwarcie pliku w celu dopisywania
@ $wp = fopen("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt", 'ab');

flock($wp, LOCK_EX);

if (!$wp) {
    echo "<p><strong> Zamówienie Państwa nie może zostać przyjęte w tej chwili.
        Proszę spróbować później.</strong></p></body></html>";
    exit;
}

fwrite($wp, $ciagwyjsciowy, strlen($ciagwyjsciowy));
flock($wp, LOCK_UN);
fclose($wp);

echo "<p>Zamówienie zapisane.</p>";
?>
</body>
</html>
```

Odczyt z pliku

Klienci Janka mogą już składać swoje zamówienia przez sieć WWW, lecz jeżeli pracownicy firmy chcą je obejrzeć, muszą otworzyć plik własnoręcznie.

Można stworzyć interfejs WWW pozwalający pracownikom na łatwe odczytywanie plików. Kod tego interfejsu został przedstawiony na listingu 2.3.

Listing 2.3. *zobaczzamowienia.php* — interfejs pozwalający pracownikom Janka na oglądanie zawartości plików

```

<?php
// utworzenie krótkich nazw zmiennych
$DOCUMENT_ROOT = $_SERVER['DOCUMENT_ROOT'];
?>
<html>
<head>
  <title>Części samochodowe Janka – zamówienia klientów</title>
</head>
<body>
<h1>Części samochodowe Janka</h1>
<h2>Zamówienia klientów</h2>
<?php

@ $wp = fopen("$DOCUMENT_ROOT/../../zamowienia/zamowienia.txt", 'rb');

if (!$wp) {
  echo "<p><strong>Brak zamówień.
  Proszę spróbować później.</strong></p>";
  exit;
}

while (!feof($wp)) {
  $zamowienie = fgets($wp, 999);
  echo $zamowienie."<br />";
}

fclose($wp);

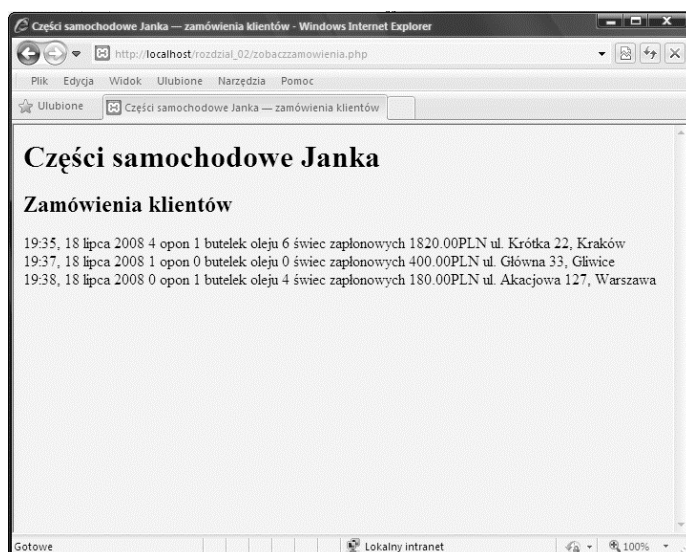
?>
</body>
</html>

```

Skrypt ten działa na zasadzie opisanej powyżej — otwarcie pliku, odczyt z pliku, zamknięcie pliku. Wynik uruchomienia powyższego skryptu, wykorzystującego plik danych z listingu 2.1, jest przedstawiony na rysunku 2.4.

Rysunek 2.4.

*Skrypt
zobaczzamowienia.php
wyświetla w oknie
przeglądarki wszystkie
zamówienia znajdujące
się w pliku
zamowienia.txt*



Należy teraz przyjrzeć się dokładnie funkcjom wykorzystanym w powyższym skrypcie.

Otwieranie pliku w celu odczytu — fopen()

Do otwarcia pliku ponownie została wykorzystana funkcja `fopen()`. W tym przypadku plik został otwarty jedynie do odczytu, tak więc zastosowano tryb `'rb'`:

```
$wp = fopen("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt", 'rb');
```

Wiedzieć, kiedy przestać — feof()

W powyższym przykładzie pętla `while` została zastosowana w celu odczytu danych z pliku aż do jego końca. Pętla `while` sprawdza koniec pliku przy użyciu funkcji `feof()`:

```
while (!feof($wp))
```

Funkcja `feof()` używa wskaźnika pliku jako swojego jedyne go parametru. Zwraca ona wartość `true`, jeżeli wskaźnik pliku znajduje się na jego końcu. Chociaż nazwa funkcji może wydawać się dziwna, łatwo ją zapamiętać, wiedząc, że `feof` znaczy w skrócie *File End Of File* (plik koniec pliku).

W tym przypadku (i zwyczajowo przy odczytywaniu pliku) plik jest odczytywany aż do osiągnięcia EOF.

Odczytywanie pliku wiersz po wierszu — fgets(), fgetss() i fgetsv()

W powyższym przykładzie do odczytania danych z pliku użyta została funkcja `fgets()`:

```
$zamowienie = fgets($wp, 999);
```

Funkcja ta jest stosowana do odczytywania pliku wiersz po wierszu. W powyższym przypadku będzie odczytywała dane, dopóki nie trafi na znak nowego wiersza (`\n`), na EOF lub przeczyta 998 bajtów pliku. Maksymalna długość odczytu jest równa wpisanej liczbie minus jeden bajt.

Istnieje wiele różnych funkcji stosowanych do odczytywania danych z pliku. Na przykład funkcja `fgets()` jest przydatna przy obróbce plików zawierających zwykły tekst, który odczytujemy fragmentami.

Interesującą odmianą `fgets()` jest funkcja `fgetss()`, która posiada następujący prototyp:

```
string fgetss(int wskaźnik_pliku, int dlugosc, string [dozwolone_znaczniiki]);
```

Funkcja ta podobna jest do `fgets()`, z wyjątkiem tego, że usuwa z czytanego ciągu wszystkie znaczniki PHP i HTML, poza wyszczególnionymi w parametrze `dozwolone_znaczniiki`. Stosuje się ją w celach bezpieczeństwa przy odczytywaniu plików napisanych przez innych programistów lub zawierających informacje wprowadzone przez użytkowników. Niekontrolowany obcy kod HTML może zniszczyć dokładnie zaplanowany proces formatowania strony. Niekontrolowany obcy kod PHP może oddać całą władzę nad serwerem złośliwemu użytkownikowi.

Inną odmianą funkcji `fgets()` jest funkcja `fgetsv()`. Posiada ona następujący prototyp:

```
array fgetsv(resource wskaźnik_pliku, int dlugosc, string [znak_podziału [, string załącznik]]);
```

Funkcja ta służy do rozdzielania wierszy pliku w celu zrekonstruowania zmiennych, kiedy wcześniej zastosowany został znak podziału (taki jak zaproponowany powyżej) lub przecinek (używany

w większości arkuszy kalkulacyjnych i innych aplikacji). Oznacza to, że przy jej stosowaniu plik jest odczytywany nie wiersz po wierszu, lecz od znaku podziału do znaku podziału. Wywołanie tej funkcji następuje podobnie jak w przypadku `fgets()`, lecz przekazuje się jej również znak podziału użyty do odseparowania pól. Na przykład:

```
$zamowienie = fgetcsv($wp, 100, "\t");
```

Powyższe polecenie odczyta wiersz z pliku i podzieli ją tam, gdzie natrafi na znak tabulacji (`\t`). Wyniki zwracane są w postaci tablicy (w powyższym przykładowym kodzie: `$zamowienie`). Tablice zostaną opisane dokładniej w rozdziale 3.

Parametr *dlugosc* powinien mieć większą wartość niż długość (wyrażoną w liczbie znaków) najdłuższego wiersza odczytywanego pliku.

Parametr *załącznik* służy do wskazywania znaków, jakimi jest otoczone każde pole w wierszu. Jeśli nie zostanie on podany, przyjęta zostanie domyślnie wartość " (podwójny cudzysłów).

Odczyt całego pliku — `readfile()`, `fpasssthru()`, `file()`

Plik można odczytywać nie tylko wiersz po wierszu, lecz również cały w jednym przebiegu. W tym celu należy posłużyć się jedną z trzech metod.

Pierwsza z nich polega na zastosowaniu funkcji `readfile()`. Można zastąpić niemal cały powyższy skrypt jednym wierszem kodu:

```
readfile("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt");
```

Wywołanie funkcji `readfile()` otwiera plik, wyświetla zawartość w oknie przeglądarki, po czym zamyka plik. Prototyp funkcji `readfile()` jest następujący:

```
int readfile(string nazwa_pliku, int [uzycie_opcji_include_path[, resource kontekst]]);
```

Opcjonalny drugi parametr określa, czy PHP powinien szukać pliku przez opcję `include_path`, i działa w sposób identyczny jak `fopen()`. Opcjonalny parametr `kontekst` używany jest jedynie wówczas, gdy pliki są otwierane zdalnie na przykład za pośrednictwem HTTP; ten sposób używania funkcji zostanie szerzej opisany w rozdziale 20. Funkcja zwraca całkowitą liczbę bajtów odczytanych z pliku.

Drugą funkcją tego typu jest `fpasssthru()`. W celu jej zastosowania należy najpierw otworzyć plik za pomocą `fopen()`, a potem przekazać wartość wskaźnika pliku funkcji `fpasssthru()`, która wyświetli zawartość tego pliku w okienku przeglądarki. Po zakończeniu działania funkcja zamyka plik.

Powyższy skrypt można zastąpić funkcją `fpasssthru()` w następujący sposób:

```
$wp = fopen("DOCUMENT_ROOT/./zamowienia/zamowienia.txt", 'rb');
fpasssthru($wp);
```

Funkcja `fpasssthru` zwraca wartość `true`, jeżeli odczyt powiedzie się, w przeciwnym wypadku — `false`.

Trzecią metodą odczytu całego pliku jest zastosowanie funkcji `file()`. Działa ona w identyczny sposób jak `readfile()` z jednym wyjątkiem — zamiast wyświetlić zawartość pliku w przeglądarce, zamienia ją na tablicę. Kwestia ta zostanie szczegółowo opisana w rozdziale 3. Tymczasem poniżej zostało przedstawione jej przykładowe zastosowanie:

```
$tablicapliku = file("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt");
```


Polecenie to wczytuje cały plik w tablicę o nazwie \$tablicaPliku. Każdy wiersz pliku zostanie zachowany jako osobny element tablicy. Warto zwrócić uwagę, że we wcześniejszych wersjach PHP funkcja `file()` nie była binarnie bezpieczna.

Czwartą dostępną opcją jest wykorzystanie funkcji `file_get_contents()`. Działa ona tak samo jak funkcja `readfile()`, z tą tylko różnicą, że zwraca zawartość pliku w postaci łańcucha znaków, a nie przesyła jej do przeglądarki.

Odczyt pojedynczego znaku — `fgetc()`

Inną metodą jest odczytywanie pliku znak po znaku. Można tego dokonać, stosując funkcję `fgetc()`. Jako jedyny parametr pobiera ona wskaźnik pliku i zwraca następny znajdujący się w pliku znak. Można zamienić pętlę `while` w przykładowym skrypcie na inną, używającą funkcji `fgetc()` w następujący sposób:

```
while (!feof($wp)) {
    $znak = fgetc($wp);
    if (!feof($wp))
        echo ($znak=="\n" ? "<br />": $znak);
}
```

Powyższy kod odczytuje za pomocą funkcji `fgetc()` pojedynczy znak z pliku i zapisuje go w zmiennej `$znak`, dopóki nie zostanie osiągnięty koniec pliku. Później zastosowana zostaje mała sztuczka zamieniająca znaki końca wiersza (`\n`), na złamania wierszy HTML (`
`).

Dzieje się tak jedynie w celu czystego sformatowania strony. W przypadku próby wyświetlenia pliku ze znakami nowych wierszy między rekordami cały plik zostałby wyświetlony jako jeden wiersz (warto sprawdzić). Przeglądarki nie generują znaków niewidocznych, dlatego trzeba je zastępować znakami HTML oznaczającymi nowy wiersz (`
`). W celu przeprowadzenia tej zamiany zastosowany został operator trójkowy.

Pomniejszym efektem ubocznym stosowania funkcji `fgetc()` zamiast `fgets()` jest fakt, że w przeciwieństwie do funkcji `fgets()` zwraca ona znak EOF. Dlatego po przeczytaniu znaku należy ponownie użyć `feof()`, aby znak ten nie został wyświetlony w przeglądarce.

Jeżeli nie istnieje wyraźny powód odczytywania pliku znak po znaku, stosowanie tej metody nie jest polecane.

Odczytywanie zadanej długości — `fread()`

Ostatnią metodą odczytywania pliku jest zastosowanie funkcji `fread()` w celu odczytania z pliku zadanej liczby bajtów. Funkcja ta posiada następujący prototyp:

```
string fread(resource wskaźnik_pliku, int dlugosc);
```

Funkcja `fread()` odczytuje przekazaną jej liczbę bajtów, chyba że wcześniej napotka znak końca pliku lub pakietu sieciowego.

Inne przydatne funkcje plikowe

Poza powyższymi istnieje jeszcze kilka przydatnych w niektórych zastosowaniach funkcji plikowych. Część z nich zostanie opisana poniżej.

Sprawdzanie istnienia pliku — file_exists()

W celu sprawdzenia istnienia pliku bez otwierania go stosuje się funkcję `file_exists()`:

```
if (file_exists("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt")) {
    echo 'Są zamówienia czekające na przyjęcie.';
} else {
    echo 'Aktualnie nie ma żadnych zamówień.';
}
```

Określanie wielkości pliku — filesize()

W celu sprawdzenia wielkości pliku stosuje się funkcję `filesize()`:

```
echo filesize("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt");
```

Funkcja ta zwraca wielkość pliku w bajtach i może zostać zastosowana w połączeniu z funkcją `fread()` w celu odczytania jednorazowo całego pliku (lub jakiejś jego części). Można zastąpić cały przykładowy skrypt następującymi wierszami kodu:

```
$wp = fopen("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt", 'rb');
echo nl2br(fread($wp, filesize("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt")));
fclose($wp);
```

Funkcja `nl2br()` przekształca w generowanym kodzie HTML znaki `\n` w znaki nowego wiersza (`
`).

Kasowanie pliku — unlink()

Można skasować plik zamówień po ich przyjęciu, stosując w tym celu funkcję `unlink()` (nie istnieje funkcja o nazwie `delete`). Na przykład:

```
unlink("$DOCUMENT_ROOT/./zamowienia/zamowienia.txt");
```

Powyższa funkcja zwraca wartość `false`, jeżeli plik nie mógł zostać usunięty. Sytuacja taka zdarza się zazwyczaj z powodu niewystarczających praw do pliku bądź jeżeli plik nie istnieje.

Poruszanie się wewnątrz pliku — rewind(), fseek() i ftell()

Można poruszać się w obrębie pliku i poznawać pozycje wskaźnika wewnątrz tego pliku, stosując funkcje `rewind()`, `fseek()` i `ftell()`.

Funkcja `rewind()` ustawia wskaźnik pliku z powrotem na jego początku. Funkcja `ftell()` informuje, jak daleko (w bajtach) został przesunięty wskaźnik. Na końcu powyższego skryptu (przed poleceniem `fclose()`) można na przykład dodać następującą wiersze :

```
echo 'Końcowa pozycja wskaźnika pliku wynosi ' . (ftell($wp));
echo '<br />';
rewind($wp);
echo 'Po przewinięciu, pozycja wynosi ' . (ftell($wp));
echo '<br />';
```

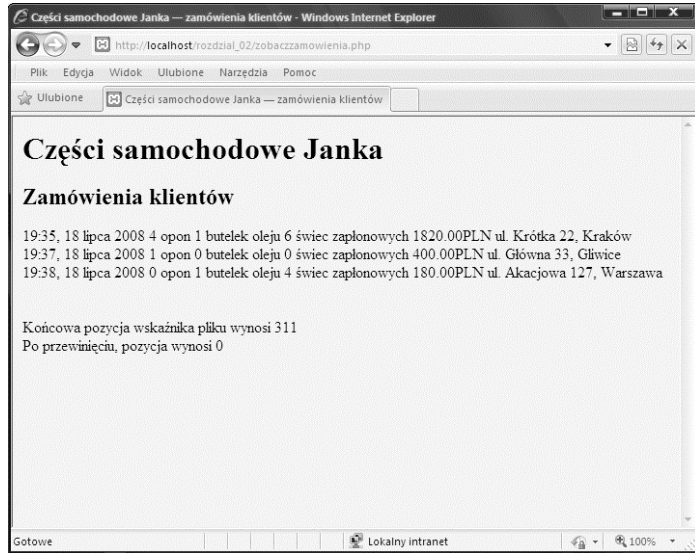
Wynik wyświetlony w przeglądarce powinien być podobny do przedstawionego na rysunku 2.5.

Funkcja `fseek()` stosowana jest do ustawiania wskaźnika pliku w dowolnym punkcie pliku. Jej prototyp wygląda następująco:

```
int fseek(resource wskaźnik_pliku, int offset [, int skąd]);
```

Rysunek 2.5.

Kiedy zamówienia są przeczytane, wskaźnik pliku wskazuje na jego koniec, offset 311 bajtów. Wywołanie przewinięcia ustawia go na pozycji 0 (na początku pliku)



Wywołanie funkcji `fseek()` ustawia wskaźnik pliku *wp* w punkcie wskazywanym przez *skąd* i przesuwa o *offset* bajtów, licząc od początku pliku. Opcjonalny parametr *skąd* posiada domyślną wartość `SEEK_SET`, czyli początek pliku. Parametr ten może również mieć wartości `SEEK_CUR` (bieżąca pozycja wskaźnika pliku) oraz `SEEK_END` (koniec pliku).

Funkcja `rewind()` jest równoznaczna z wywołaniem funkcji `fseek()` z zerowym offsetem. Na przykład można zastosować funkcję `fseek()` w celu znalezienia środkowego rekordu w pliku danych lub po to, aby przeprowadzić przeszukiwanie binarne. Zazwyczaj po osiągnięciu poziomu złożoności wymagającego stosowania takich mechanizmów polecane jest zastosowanie bazy danych.

Blokowanie pliku

Można wyobrazić sobie sytuację, w której dwóch klientów stara się zamówić produkt w tym samym czasie (dzieje się tak często, zwłaszcza gdy strona jest licznie odwiedzana). Co się stanie, gdy jeden z klientów wywoła funkcję `fopen()` i zacznie zapis w pliku, a drugi uczyni to samo? Jak będzie wyglądać ostateczna zawartość pliku? Czy najpierw zostanie zapisane pierwsze zamówienie, a później drugie, czy też odwrotnie? A może dojdzie do sytuacji niepożądaney, na przykład oba zamówienia wymieszają się ze sobą? Odpowiedź na powyższe pytania zależy od konkretnego systemu operacyjnego, ale zazwyczaj jest to wielka niewiadoma.

W celu uniknięcia powyższych problemów stosuje się blokowanie plików. W PHP wykorzystuje się w tym celu funkcję `flock()`. Powinna ona zostać wywołana po otwarciu pliku, lecz przed odczytaniem go lub zapisaniem w nim danych.

Funkcja `flock()` posiada następujący prototyp:

```
bool flock(resource wskaźnik_pliku, int dzialanie [, int zablokuj])
```

Funkcji `flock()` należy przekazać wskaźnik otwartego pliku i cyfrę określającą wymagany rodzaj zamka. Funkcja ta zwraca `true`, jeżeli zamek został prawidłowo założony, a `false` w przeciwnym wypadku. Opcjonalny trzeci parametr będzie zawierał wartość `true`, jeśli założenie zamka spowoduje zablokowanie bieżącego procesu (czyli zmuszenie go do przejścia w stan oczekiwania).

Możliwe wartości parametru *działanie* są przedstawione w tabeli 2.2.

Tabela 2.2. Wartości parametru *działanie* funkcji *flock()*

Wartość parametru <i>działanie</i>	Znaczenie
LOCK_SH (dawniej 1)	Blokowanie odczytu. Pozwala na dzielenie pliku z innymi czytającymi
LOCK_EX (dawniej 2)	Blokowanie zapisu. Wyłącza plik z użytku; nie może on być dzielony
LOCK_UN (dawniej 3)	Zwolnienie istniejącej blokady
LOCK_NB (dawniej 4)	Dodanie wartości 4 do parametru <i>działanie</i> przeciwdziała zablokowaniu próby założenia blokady

Stosując funkcję `flock()`, należy dodać ją do wszystkich skryptów korzystających z pliku. W innym przypadku jest ona bezwartościowa.

Należy zwrócić uwagę, iż `flock()` nie działa w systemie NFS i innych sieciowych systemach plików. Nie działa ona również w starszych systemach plików, które nie obsługują blokowania plików — systemem takim jest na przykład FAT. W niektórych systemach operacyjnych funkcja ta jest zaimplementowana na poziomie procesu i nie będzie działać prawidłowo, jeśli stosowany będzie wielowątkowy API serwera.

Aby zastosować ją w powyższym przykładzie, należy zmienić skrypt *przetwarzamowienie.php* w następujący sposób:

```
$wp = fopen("$DOCUMENT_ROOT../zamowienia/zamowienia.txt", 'ab');
flock($wp, LOCK_EX); // blokada zapisu pliku
fwrite($wp, $ciagwyjsciowy);
flock($wp, LOCK_UN); // zwolnienie blokady zapisu
fclose($wp);
```

Do skryptu *zobaczamowienia.php* należy również dodać blokady:

```
$wp = fopen("$DOCUMENT_ROOT../zamowienia/zamowienia.txt", 'r');
flock($wp, LOCK_SH); // blokada odczytu pliku
// odczyt z pliku
flock($wp, LOCK_UN); // zwolnienie blokady odczytu
fclose($wp);
```

Kod jest teraz dużo solidniejszy, ale ciągle niedoskonały. Co by się stało, gdyby dwa skrypty jednocześnie usiłowały założyć blokadę? Spowodowałoby to „wyścig” o bardzo niepewnym wyniku, co wywołałoby wiele kolejnych problemów. Lepszą metodą jest zastosowanie *DBMS* (ang. *Database Management System* — system zarządzania bazami danych).

Lepszy sposób obróbki danych — systemy zarządzania bazami danych

Wszystkie powyższe przykłady używały plików jednorodnych. W drugiej części tej książki opiszę alternatywę tej metody — MySQL, system zarządzania relacyjnymi bazami danych (RDBMS). Można by zapytać, w jakim celu?

Problemy związane ze stosowaniem plików jednorodnych

Istnieje kilka problemów związanych z pracą z plikami jednorodnymi:

- Praca z dużym plikiem może być bardzo powolna.
- Poszukiwanie konkretnego rekordu lub grupy rekordów w pliku jednorodnym jest trudne. Jeżeli rekordy są uporządkowane, można zastosować pewien sposób przeszukiwania binarnego w połączeniu z rekordami o ustalonej szerokości i przeszukiwaniem według pola kluczowego. Aby znaleźć pewne wzory informacji (na przykład wyszukując wszystkich klientów zamieszkałych w Gliwicach), należy sprawdzać indywidualnie każdy rekord.
- Problemy sprawiają dostęp jednoczesny. Powyżej przedstawione zostały sposoby blokowania plików, ale, jak opisano powyżej, może to spowodować wyścig lub „wąskie gardło”. W razie dużego ruchu na stronie liczna grupa użytkowników może czekać na odblokowanie pliku, aby złożyć zamówienie. Jeżeli potrwa to zbyt długo, przeniosą się do konkurencji.
- Wszystkie przedstawione powyżej procesy przetwarzania plików działają sekwencyjnie — rozpoczynają od początku pliku i czytają go do końca. Aby umieścić lub skasować rekordy znajdujące się w środku pliku, należy umieścić cały plik w pamięci, dokonać zmian, a na końcu zapisać go w całości. Podczas pracy z dużymi plikami konieczność wykonywania wszystkich tych kroków może sprawiać problemy.
- Poza ograniczonymi możliwościami, oferowanymi przez pozwolenia dostępu do plików, nie istnieje żadna prosta metoda tworzenia różnych poziomów dostępu do danych.

Jak RDBMS rozwiązują powyższe problemy?

Relacyjne systemy zarządzania bazami danych umożliwiają rozwiązania wszystkich powyższych kwestii.

- RDBMS pozwalają na znacznie szybszy dostęp do plików niż pliki jednorodne. MySQL, system bazodanowy prezentowany w tej książce, należy do najszybszych RDBMS.
- RDBMS można zadawać zapytania o dane spełniające konkretne kryteria.
- RDBMS posiadają wbudowany mechanizm zapewniania równoległego dostępu, który pozostaje poza kręgiem pracy programisty.
- RDBMS pozwalają na swobodny dostęp do danych.
- RDBMS posiadają wbudowany system przywilejów. MySQL jest w tej dziedzinie szczególnie rozbudowany.

Prawdopodobnie głównym powodem używania RDBMS jest fakt, że funkcjonalność, którą powinny posiadać systemy przechowywania danych, została już w nich zaimplementowana (a przynajmniej jej większość). Oczywiście można napisać własną bibliotekę funkcji PHP, lecz po co ponownie wymyślać koło?

W części II tej książki opiszemy ogólną zasadę działania relacyjnych baz danych, a w szczególności konfigurację i zastosowanie MySQL w tworzeniu stron WWW opartych na bazach danych.

Jeżeli tworzony jest prosty system, który nie wymaga pełnowymiarowej bazy danych, natomiast chcemy uniknąć zakładania zamków i innych komplikacji związanych z używaniem plików płaskich, można użyć nowego rozszerzenia PHP o nazwie SQLite. Udostępnia ono SQL-owy interfejs do pliku płaskiego. W tej książce skupimy się na używaniu serwera MySQL, natomiast więcej informacji na temat SQLite można znaleźć pod adresami <http://sqlite.org/> oraz <http://www.php.net/sqlite>.

Propozycje dalszych lektur

Więcej informacji na temat interakcji z systemem plików znajduje się w rozdziale 19. Opiszemy w nim metody zmiany pozwoleń dostępu, własności i nazw plików, a także pracę z katalogami oraz interakcję ze środowiskiem systemu plików.

Zalecamy również lekturę rozdziału na temat systemów plików w podręczniku elektronicznym PHP, dostępnym pod adresem <http://www.php.net/filesystem>.

W następnym rozdziale

W kolejnym rozdziale przedstawimy tablice — czym są i jak mogą zostać zastosowane w skryptach PHP do przetwarzania danych.

Skorowidz

-, 55
--, 57
' , 359, 493
!, 60
!=, 59, 62, 103
!==, 59, 62, 103
", 493
#, 45
\$, 47
\$_COOKIE, 54, 480
\$_ENV, 54
\$_FILES, 54, 414
\$_GET, 48, 54
\$_POST, 47, 48, 54, 263
\$_REQUEST, 48, 54, 467, 609
\$_SERVER, 54, 183, 407
\$_SESSION, 54, 479, 482, 483
\$DOCUMENT_ROOT, 80
\$GLOBALS, 54
\$HTTP_POST_VARS, 47, 489
\$HTTP_SESSION_VARS, 485
\$this, 170
%, 55
%=: 57
&, 58, 60
&&, 60
";, 355
*, 55
*=: 57
,, 60
., 46, 49
.=, 57
/, 55
/* */ , 45
//, 45
/=: 57
::, 176, 186
?., 60
@, 61, 203, 264
[], 61
\\, 125
^, 60
__autoload(), 189
__call(), 188, 189
__clone(), 188
__construct(), 168, 171
__destruct(), 169, 171
__get(), 170, 180
__set(), 170, 180
__toString(), 191, 192
`, 359
`` , 61, 426
{ }, 68
|, 60
||, 60
~, 60
+, 55, 62, 103
++, 57
+=: 57
<, 59
<% %>, 44
<?php ?>, 43, 499
<<, 60
<<<, 50
<=: 59
<>, 59, 62, 103

, 91, 122
<form>, 40, 413
 action, 41
 method, 41
<html>, 48
, 464
<script>, 44
<table>, 154
=, 51, 55, 56
=: 57
==, 58, 59, 62, 103
===, 59, 62, 103, 160
=>, 61
>, 59
>=: 60, 170, 173
>=: 59
>>, 60
l-Click, 318

A

- a2ps, 701
- abort(), 769
- abstract, 188
- ACID, 299
- Acrobat Reader, 715
- action, 41
- Active Server Pages, 44
- ADD INDEX, 291
- ADDDATE(), 454
- addslashes(), 125, 263, 270, 284, 398, 494, 507
- administracja MySQL, 275
- Adobe Distiller, 700
- Adobe PDF, 699
- AdobeAcrobat Reader, 699
- adres
 - e-mail, 127, 138, 543
 - IP, 373, 435
 - URL, 380, 430, 435, 481, 764, 784
- Advanced Maryland Automated Network
 - Disk Archiver, 342
- agregowanie danych, 250
- Ajax, 763, 764, 768
 - aplikacje, 768
 - asynchroniczny tryb połączeń, 765
 - biblioteki JavaScript, 789
 - CSS, 765, 766
 - DHTML, 765
 - DOM, 789
 - getElementsByTagName(), 772
 - JavaScript, 767
 - kaskadowe arkusze stylów, 766
 - komunikacja z serwerem, 770
 - przetwarzanie odpowiedzi serwera, 772
 - readyState, 772
 - skrypty działające po stronie klienta, 767
 - skrypty JavaScript, 765
 - tworzenie obiektu XMLHttpRequest, 770
 - witryny internetowe, 790
 - XHTML, 766
 - XML, 765, 768
 - XMLHttpRequest, 768, 770
 - ZakładkaPHP, 777
 - żądania HTTP, 764
- Ajax Lines, 790
- Ajax Matters, 790
- ajaxCzasSerwera.html, 773
- Ajaxian, 790
- akcje formularza, 41
- aktualizacja witryny, 314
- Aktywne Strony Serwera, 44
- algorytm szyfrujący, 335, 336
 - DES, 337
 - RSA, 337
- aliasy, 248
- ALIKE, 244
- ALL, 225, 253
- ALL PRIVILEGES, 225
- allow_url_fopen, 82, 357
- ALTER, 223
- ALTER TABLE, 255, 256, 299, 300, 301
 - ADD, 256
 - ADD FOREIGN KEY, 256
 - ADD INDEX, 256, 291
 - ADD PRIMARY KEY, 256
 - ADD UNIQUE, 256
 - ALTER, 256
 - CHANGE, 256
 - CHARACTER SET, 256
 - CONVERT TO CHARACTER SET, 256
 - DISABLE KEYS, 256
 - DISCARD TABLESPACE, 256
 - DROP, 256
 - DROP FOREIGN KEY, 256
 - DROP INDEX, 256
 - DROP PRIMARY KEY, 256
 - ENABLE KEYS, 256
 - IMPORT TABLESPACE, 256
 - MODIFY, 256
 - ORDER BY, 256
 - RENAME, 256
- alternatywna składnia struktur sterujących, 75
- AMANDA, 342
- Amazon.com, 318, 725, 729, 730
 - interfejs usług sieciowych, 730
 - węzły przeglądania, 732
- analiza integralności plików, 330
- analiza klas i obiektów, 192
- analiza wizyt na serwerze, 314
- analiza wykorzystania kluczy, 290
- analizator składniowy, 518
- Analog, 314
- ANALYZE TABLE, 290
- and, 60
- AND, 244
- anomalie
 - modyfikacje, 212
 - uaktualnianie, 211
 - usuwanie, 213
 - wstawianie, 212
- anonimowe logowanie, 438
- ANSI SQL, 240, 251
- anti-aliasing, 462
- ANY, 253
- Apache, 339, 361, 363
 - Auth_MySQL_Info, 387
 - AuthGroupFile, 385
 - AuthName, 385
 - AuthType, 385
 - AuthUserFile, 384
 - certyfikaty, 799
 - CGI, 802
 - dodawanie obsługi PHP, 807

- ErrorDocument, 384
- httpd.conf, 800
- instalacja, 804
- ISAPI, 802
- konfiguracja, 363
- mod_auth, 383
- mod_auth_mysql, 386
- OpenSSL, 798
- pliki .htaccess, 384
- podstawowe uwierzytelnianie, 383
- sprawdzanie obsługi PHP, 800
- SSL, 800, 801
- testowanie obsługi SSL, 801
- API Reflection, 192
- aplikacje
 - Ajax, 764, 768
 - PHP, 40
 - tworzące PDF, 702
 - Web 2.0, 763, 764
 - WWW, 503
- aplikacje internetowe, 503
 - aplikacje działające na serwerach komercyjnych, 364
 - bezpieczeństwo, 345
 - użyteczność, 346
- apostrofy, 355
- apostrofy wykonywania poleceń systemu operacyjnego, 359
- Apple LaserWriter, 701
- architektura internetowej bazy danych, 215
- ARCHIVE, 298
- archiwa aplikacji PHP, 33
- argumenty, 46
- arkusze stylów, 766
- Array, 52
- array(), 98, 100, 609
- array_count_values(), 116
- array_pop(), 111
- array_push(), 111
- array_reverse(), 110, 111, 112
- array_walk(), 115
- arsort(), 107
- AS, 248
- ASC, 232
- ASCII, 237, 696
- ASIN, 729
- asort(), 107
- ASP, 44, 768
- ASP.NET, 30, 44
- asp_tags, 44
- AssociateTag, 756
- asynchroniczny tryb połączeń, 765
- Asynchronous JavaScript and XML, 763
- ataki
 - Cross Site Scripting, 349
 - DDoS, 330, 348, 369
 - DoS, 330, 348, 369
 - XSS, 349

- Atomicity, 299
- atomowe wartości kolumn, 213
- atomowość, 299
- atrybuty, 166, 167, 168, 170, 208
- atrybuty kolumn, 228
- atrybuty pliku, 424
- Auth_MySQL_DB, 388
- Auth_MySQL_Encryption_Types, 388
- Auth_MySQL_Info, 387
- Auth_MySQL_Password_Field, 388
- Auth_MySQL_Password_Table, 388
- Auth_MySQL_Username_Field, 388
- AuthGroupFile, 385
- AuthName, 385
- AuthType, 385
- AuthUserFile, 384
- auto_append_file, 148, 380
- AUTO_INCREMENT, 228, 229, 298
- auto_prepend_file, 148, 380
- AUTOCOMMIT, 300
- automatycznie generowane obrazki, 464
- autorzy systemów, 350
- AVG(), 250
- awaria, 371
- awarie sprzętu komputerowego, 322
- AWS, 731
- AWS Developer Guide, 731
- AWSAccessKeyId, 756
- AWSZone, 731

B

- Backflip, 533
- background, 766
- badanie reprezentatywnych grup użytkowników, 314
- basename(), 417, 421, 423
- baza danych, 207
 - anomalie modyfikacji, 212
 - anomalie uaktualniania, 211
 - anomalie usuwania, 213
 - anomalie wstawiania, 212
 - atomowe wartości kolumn, 213
 - atrybuty, 208
 - bezpieczeństwo, 282
 - funkcje składowane, 303
 - indeksy, 232, 292
 - informacje, 284
 - informacje o kolumnach, 286
 - instrukcje przygotowywane, 270
 - klient GUI, 218
 - klucz podstawowy, 209
 - klucze, 209, 214
 - klucze obce, 209, 301
 - kolumny, 208
 - kopia zapasowa, 292
 - krotki, 208
 - kursory, 305

- baza danych
 - MDB2, 272
 - MySQL, 29, 217
 - ODBC, 31
 - odczytywanie rezultatów zapytań, 266
 - optymalizacja, 291
 - optymalizacja projektu, 291
 - optymalizacja tabel, 291
 - pola, 208
 - procedury składowane, 302
 - projektowanie, 207, 211
 - przywracanie bazy danych, 293
 - RDBMS, 94
 - redundancja danych, 211
 - rekordy, 208
 - relacje, 210
 - replikacja, 293
 - schemat, 209
 - SQL, 239
 - struktury sterujące, 305
 - system uprawnień, 365
 - tabele, 208
 - transakcje, 215, 299
 - tworzenie, 217
 - tworzenie tabel, 227
 - ustanawianie połączenia, 264
 - usuwanie bazy, 258
 - usuwanie rekordów, 257
 - usuwanie tabel, 257
 - użytkownicy, 365
 - wartości, 208
 - wartości domyślne, 292
 - wstawianie danych, 267
 - wybór bazy, 265
 - wysyłanie danych do serwera, 366
 - wyszukiwanie danych, 242
 - zamykanie połączenia, 267
 - zapisywanie danych, 240
 - zapytania, 214
 - zmiana rekordów, 255
 - zmiana struktury tabel, 255
- bazowa linia pisma, 470
- BEGIN, 302, 303
- BETWEEN, 244
- bezpieczeństwo, 325, 333, 345
 - baza danych, 282
 - hasła, 282
 - internet, 283
 - numery kart kredytowych, 400
 - przechowywanie danych, 399
 - przywileje użytkowników, 283
 - serwer bazy danych, 365
 - skrypty, 48
 - SSL, 284
 - system operacyjny, 282
 - transakcje, 391
 - uwierzytelnianie, 334, 373
 - bezpieczeństwo aplikacji internetowych, 345
 - apostrofy wykonywania poleceń systemu operacyjnego, 359
 - ataki DDoS, 348, 369
 - ataki DoS, 348, 369
 - autorzy systemów, 350
 - baza danych, 365
 - błędy, 358
 - dostęp do danych poufnych, 347
 - faza od ogółu do szczegółu, 347
 - faza od szczegółu do ogółu, 347
 - filtrowanie danych pochodzących od użytkowników, 351
 - fizyczne zabezpieczenie serwera, 370
 - identyfikacja użytkowników, 349
 - instalacja nowych wersji oprogramowania, 361
 - kod źródłowy, 351
 - komputer, 369
 - konfiguracja serwera WWW, 362
 - monitorowanie bezpieczeństwa, 347
 - nieświadomi użytkownicy zainfekowanych komputerów, 350
 - organizacja kodu źródłowego, 356
 - PHP, 360
 - php.ini, 362
 - planowanie działań na wypadek awarii, 371
 - planowanie z wyprzedzeniem, 346
 - połączenie z serwerem, 366
 - rozczarowani pracownicy, 350
 - rozpoznawanie zagrożeń, 347
 - serwer bazy danych, 365
 - serwer WWW, 360
 - sieć, 367
 - sprawdzanie obecności kodu SQL w ciągach znaków, 354
 - stabilność kodu, 358
 - strategie zapewniania bezpieczeństwa, 345
 - strefa zdemilitaryzowana, 368
 - system operacyjny, 369
 - system plików, 358
 - uaktualnianie oprogramowania, 361
 - uaktualnianie systemu operacyjnego, 369
 - udostępnianie tylko niezbędnych usług, 370
 - unieważnianie danych wynikowych, 355
 - utrata danych, 348
 - użyteczność, 346
 - wdrażanie nowych wersji, 361
 - wstrzykiwanie kodu, 349
 - wymagany poziom bezpieczeństwa, 347
 - wysyłanie danych do serwera, 366
 - XSS, 349
 - zablokowanie usługi, 348
 - zapora sieciowa, 368
 - zawartość kodu źródłowego, 357
 - złamanie zabezpieczeń dostępu do serwera, 349
 - złodziejce sprzętu komputerowego, 350
 - zniszczenie danych, 348

- bezpieczeństwo fizyczne, 343
 - pożar, 343
 - UPS, 343
 - bezpieczeństwo komercyjnych stron WWW, 325
 - blokada usługi, 330
 - błędy w oprogramowaniu, 331
 - certyfikaty cyfrowe, 339
 - hasła, 334
 - informacje, 326
 - kopia zapasowa, 342
 - modyfikacje danych, 329
 - monitorowanie zdarzeń, 340
 - ochrona plików, 329
 - podpis cyfrowy, 338
 - polityka bezpieczeństwa, 333
 - serwer WWW, 339
 - SSL, 340
 - szyfrowanie, 335
 - testowanie, 332
 - ujawnienie informacji poufnych, 327
 - utrata danych, 328
 - wadliwa dokumentacja, 331
 - zagrożenia bezpieczeństwa, 326
 - zapisywanie zdarzeń, 340
 - zapora sieciowa, 341
 - zaprzeczenie korzystania z usługi, 332
 - zasady uwierzytelniania, 334
 - zniszczenie danych, 328
 - bezpieczne serwery WWW, 339
 - biblioteki, 31
 - PECL, 806
 - BIGINT, 234
 - BINARY, 237
 - binary large objects, 234
 - bind_param(), 271
 - bind_result(), 271
 - BIT, 234
 - Bitkeeper, 511
 - BLOB, 234, 236, 237, 298
 - blok catch, 195
 - blok kodu, 68
 - blok try, 195
 - blokada usługi, 330
 - blokowanie plików, 93
 - błędy, 151, 358
 - @, 203
 - komunikaty, 518
 - obsługa, 202, 529
 - poziomy zgłaszania, 527
 - programistyczne, 517
 - stałe zgłaszania błędów, 527
 - stos wywołań, 197
 - wywołanie nieistniejących funkcji, 520
 - wyzwalanie błędów, 529
 - zmiana ustawień zgłaszania, 528
 - błędy analizatora składniowego, 517
 - błędy kompilacja, 517
 - błędy krytyczne, 527
 - błędy logiczne, 523
 - błędy składniowe, 517
 - błędy w oprogramowaniu, 331
 - błędy wykonania, 519, 523
 - Bogaty Format Tekstu, 695
 - BOOL, 234
 - Boolean, 51
 - border, 154
 - bounding box, 469
 - break, 70, 75
 - broszury internetowe, 312
 - aktualizacja witryny, 314
 - informacje, 312
 - jakość prezentacji, 313
 - zainteresowanie efektami działania witryny WWW, 314
 - zgłoszenia potencjalnych klientów, 313
 - browse nodes, 732
 - buforowanie, 732, 756
 - BUGTRAQ, 416
- ## C
- calendar, 456
 - CALL, 303
 - Cascading Style Sheets, 144, 513
 - case, 69
 - CASE, 307
 - catch, 195
 - cellpadding, 154, 155
 - cellspacing, 154, 155
 - centrum certyfikacji, 339
 - Certificate Signing Request, 340
 - Certifying Authorities, 339
 - certyfikaty autentyczności, 333
 - certyfikaty cyfrowe, 339, 799
 - certyfikaty PDF, 699
 - CGI, 395, 428, 802
 - CHAR, 234, 237
 - checkdate(), 449
 - checkdnsrr(), 436
 - chgrp(), 424
 - chmod, 82
 - chmod(), 424
 - chop(), 121
 - chown(), 424
 - ciagi znaków, 50, 51
 - cudzysłowy, 50
 - cudzysłowy pojedyncze, 50
 - definiowanie, 50
 - długość, 130
 - dopasowywanie podciągów, 130
 - dostęp do podciagu, 128
 - formatowanie, 121, 122
 - formatowanie do przechowania, 125
 - formatowanie do wyświetlenia, 122
 - formatowanie HTML, 122

- ciągi znaków
 - łączenie, 49, 127
 - odnajdywanie pozycji podciągu, 131
 - operatory, 56
 - porównywanie, 129
 - porządkowanie, 129
 - przeszukiwanie, 131
 - przetwarzanie, 119
 - przycinanie, 121
 - rozdzielanie, 127
 - składnia heredoc, 50
 - sprawdzanie długości, 130
 - wyrażenia regularne, 133
 - zamiana podciągów, 130, 132
 - zmiana wielkości liter, 125
 - ciphertext, 335
 - class, 168, 179
 - clearstatcache(), 424
 - ClibPDF, 702
 - clone, 188
 - CLOSE, 307
 - close(), 267, 420
 - closedir(), 419
 - Cold Fusion, 30
 - columns_priv, 276, 278, 281
 - COMMIT, 298, 300
 - Concurrent Versions System, 510
 - Consistency, 299
 - const, 186
 - Content-type, 463
 - continue, 75
 - CONTINUE, 306
 - cookie, 54, 480
 - konfiguracja, 480
 - sesje, 481
 - tworzenie, 480
 - ustawianie, 480
 - copy(), 425
 - cos(), 723
 - count(), 112, 116
 - COUNT(), 250
 - cracker, 321, 350
 - CREATE, 223
 - CREATE DATABASE, 220
 - CREATE INDEX, 232
 - CREATE PROCEDURE, 302
 - CREATE TABLE, 227, 230, 298
 - CREATE TEMPORARY TABLES, 224
 - CROSS JOIN, 245
 - Cross Site Scripting, 349
 - crypt(), 378
 - Crypt_DES, 388
 - CSR, 340
 - CSS, 144, 513, 764, 765, 766
 - deklaracje, 766
 - klasy, 767
 - selektory, 766
 - wartości, 766
 - CSV, 90, 298
 - cudzysłowy, 50, 125, 355, 493
 - cudzysłowy podwójne, 50
 - cURL, 393, 432
 - szyfrowanie, 393
 - current(), 114
 - CVS, 510, 511, 516
 - czas, 46, 234, 445
 - mikrosekundy, 455
 - czas modyfikacji pliku, 440
 - czasowa zmiana środowiska wykonawczego, 497
 - czasserwera.php, 771
 - czcionki, 462, 465, 468
 - PostScript Type 1, 458
 - TrueType, 458, 465
- ## D
- dane, 77, 399
 - dane osobiste użytkownika, 317, 374
 - dane poufne, 347
 - dane z witryn WWW, 430
 - data, 46, 234, 445
 - kalendarze, 455
 - konwersja pomiędzy formatami daty PHP i MySQL, 450
 - MySQL, 450
 - obliczenia, 453, 454
 - sprawdzanie poprawności, 449
 - znacznik czasu Uniksa, 447
 - Data Definition Language, 240
 - Data Encryption Standard, 337
 - Data Manipulation Language, 240
 - DATE, 236
 - date(), 46, 445, 518, 771
 - kody formatów, 446
 - znacznik czasu Uniksa, 447
 - date_add(), 454
 - date_diff(), 454
 - DATE_FORMAT(), 450, 452
 - kody formatów, 452
 - date_sub(), 454
 - DATEDIFF(), 455
 - DATETIME, 236
 - daty, 353
 - db, 276, 278
 - DDL, 240, 301
 - DDoS, 330, 348, 369
 - DEC, 235
 - DECIMAL, 235
 - declare, 76
 - DECLARE, 304
 - DECLARE CONTINUE HANDLER, 306
 - decoct(), 424
 - default, 71
 - DEFAULT, 304
 - define(), 53

- definicja klasy, 168
- definicja typu dokumentu, 697
- Definicja Typu Dokumentu, 728
- definiowanie
 - ciągi, 50
 - funkcje, 152
 - interfejs, 178
 - konwencje nazewnictwa, 506
 - przywileje, 220
 - stałe, 53
 - użytkownicy, 220
- deklaracja
 - funkcje, 152
 - funkcje składowane, 303
 - podprogramy obsługi, 306
 - procedury składowane, 302
 - stałe, 53
- dekrementacja, 57
- DELAYED, 241
- DELETE, 223, 226, 257
- DELIMITER, 302, 303
- demilitarized zone, 350, 368
- Denial of Service, 330, 348
- DES, 336, 337, 388
- DESC, 232
- DESCRIBE, 231, 284, 286
- deserializacja, 758
- deserialize(), 758
- destruktor, 169
- deszyfracja, 336
- DHTML, 765
- diagramy encji i relacji, 209
- die(), 495
- dir, 420
- dirname(), 421, 423
- disaster planning, 371
- disk_free_space(), 421
- display_errors, 528, 798
- DISTINCT, 250
- Distributed Denial of Service, 330, 348
- długość ciągu znaków, 130
- DML, 240
- DMZ, 350, 368
- DNS, 435, 436
- dns_get_mx(), 434, 436
- do..while, 74
- doc_root, 807
- Document Object Model, 765, 789
- Document Type Definition, 728
- DOCUMENT_ROOT, 80
- dodawanie PHP do konfiguracji serwera Apache, 807
- dodawanie zawartości dynamicznej, 45
- Dojo, 790
- Dokument Type Definition, 697
- dokumentacja, 331, 510
- dokumentacja projektów, 511
 - phpautodoc, 512
 - phpdoc, 512
 - PHPDocumentor, 512
- dokumenty, 696
 - ASCII, 696
 - DTD, 728
 - PostScript, 698
 - RFC, 429, 443
 - WSDL, 730
- dokumenty PDF, 695, 699
 - czcionki, 715
 - generowanie, 713
 - hiperłącza, 715
 - kodowanie znaków, 716
 - liczba stron, 715
 - rozmiar strony, 715
 - tworzenie, 700
 - tworzenie z szablonu, 710
 - wysyłanie do przeglądarki, 717
 - zakładki, 715
- dokumenty RTF, 695, 697
 - tworzenie, 707
- dokumenty XML, 43, 727
 - wczytywanie, 751
- dołączanie plików, 142, 356
- DOM, 765, 789
- domeny najwyższego poziomu, 138
- dopasowanie podciągów, 130
- dopasowanie tekstu do przycisku, 468
- DoS, 330, 369
- dostarczanie usług lub wyrobów mających
 - postać cyfrową, 319
- dostęp do danych za pośrednictwem zmiennych, 47
- dostęp do elementów tablicy, 61, 101
- dostęp do składowych klasy, 60
- dostęp do stałej klasowej, 186
- dostęp do zawartości tablicy, 99
- dostęp do zmiennych formularza, 47
- dostosowywanie wyrażeń regularnych, 133
- Double, 51
- DOUBLE, 235
- DOUBLE PRECISION, 235
- doubleval(), 270
- DROP, 223
- DROP DATABASE, 258
- DROP FUNCTION, 304
- DROP PROCEDURE, 304
- DROP TABLE, 258
- drukarka PostScript, 701
- drzewa, 672
- DSO, 798
- DTD, 697, 728
- Durability, 299
- duże obiekty binarne, 234
- duże projekty, 503
- dvips, 701
- dynamic shared objects, 798
- dynamiczne witryny WWW, 28
- Dynamiczny HTML, 765
- dyrektywy wykonania, 76

dziedziczenie, 167, 173
 final, 176
 implementacja, 173
 interfejsy, 177
 klasy pochodne, 167
 kontrola widoczności, 174
 modyfikatory dostępu, 174
 podklasy, 167
 potomek, 167
 przodek, 167
 unieważnianie, 175
 wielodziedziczenie, 177
 wywołanie operacji z klasy przodka, 176
 zapobieganie dziedziczeniu, 176

dzielenie kodu, 509
 dzielenie wewnętrznych funkcji, 510

E

E_ALL, 527
 E_COMPILE_ERROR, 527
 E_COMPILE_WARNING, 527
 E_CORE_ERROR, 527
 E_CORE_WARNING, 527
 E_ERROR, 527
 E_NOTICE, 527
 E_PARSE, 527
 E_RECOVERABLE_ERROR, 527
 E_STRICT, 527
 E_USER_ERROR, 527, 531
 E_USER_NOTICE, 527
 E_USER_WARNING, 527
 E_WARNING, 527
 each(), 101, 114, 115
 eBay, 729
 echo, 44, 65, 499, 514
 e-commerce, 311, 321
 edytor tekstu, 697
 egzemplarzowanie, 169
 elementy HTML, 355
 else, 68
 ELSE, 307
 elseif, 69
 empty(), 66, 121, 483
 enctype, 413
 END, 302, 303
 END CASE, 307
 END LOOP, 306
 END REPEAT, 306
 END WHILE, 306
 end(), 114
 endforeach, 75
 endif, 75
 endor, 75
 endswitch, 75
 endwhile, 75
 ENT_COMPAT, 355

ENT_NOQUOTES, 355
 ENT_QUOTES, 355
 ENUM, 235, 237
 EOF, 89, 91
 equi-join, 246, 247, 249
 ereg_replace(), 139
 eregi_replace(), 139
 error_reporting, 528
 error_reporting(), 529
 ErrorDocument, 384
 escapeshellcmd(), 360, 398, 427
 eval(), 494
 Excel, 298
 Exception, 195, 196, 197
 __toString(), 197
 getCode(), 197
 getFile(), 197
 getLine(), 197
 getMessage(), 197
 getTrace(), 197
 getTraceAsString(), 197
 exec(), 359, 425
 execute(), 271
 EXISTS, 254
 exit, 75
 EXIT, 306
 exit(), 495
 exmlm, 670
 EXPLAIN, 284, 286, 287, 288, 290, 514
 typy połączeń, 289
 explode(), 113, 127, 128, 130, 436
 extends, 173
 Extensible Hypertext Markup Language, 766
 Extensible Markup Language, 43, 726
 extension_dir, 806
 EXTR_IF_EXISTS, 118
 EXTR_OVERWRITE, 117, 118
 EXTR_PREFIX_ALL, 118
 EXTR_PREFIX_IF_EXISTS, 118
 EXTR_PREFIX_INVALID, 118
 EXTR_PREFIX_SAME, 118
 EXTR_REFS, 118
 EXTR_SKIP, 118
 extract(), 117, 118
 Extreme Programming, 516

F

Facebook, 764
 false, 51, 58, 73
 FAT, 94
 fclose(), 84, 86, 442
 FDF, 711
 fdf_create(), 711
 fdf_set_file(), 711
 fdf_set_value(), 711
 feof(), 89, 91

- FETCH, 306
- fetch(), 271
- fetch_assoc(), 266
- fetch_object(), 267
- fetch_row(), 266
- fetchRow(), 274
- fgetc(), 91
- fgetcsv(), 89, 90
- fgets(), 89, 553
- fgetss(), 89
- figury, 471
- FILE, 224, 283
- File End Of File, 89
- File Transfer Protocol, 429
- file(), 90, 112
- file_exists(), 92, 440
- file_get_contents(), 84, 91, 432
- file_put_contents(), 84
- file_uploads, 412
- fileatime(), 423
- filegroup(), 423, 424
- fileinfo, 33
- filemtime(), 423, 440
- fileopen(), 432
- fileowner(), 423, 424
- fileperms(), 424
- filesize(), 92, 424
- filetype(), 424
- filtrowanie danych pochodzących od użytkowników, 351
 - filtrowanie wartości, 353
 - kod SQL w ciągach znaków, 354
 - podwójne sprawdzanie oczekiwanych wartości, 351
- filtrowanie ruchu sieciowego, 368
- filtrowanie wartości, 353
- final, 176
- firewall, 282, 341
- FishCartSQL, 598
- FIXED, 235
- Flickr, 764
- float, 270
- Float, 51
- FLOAT, 235
- floatval(), 67
- flock(), 93
- FLUSH PRIVILEGES, 281
- FLUSH TABLES, 292
- FLUSH TABLES WITH READ LOCK, 295
- font-family, 766
- font-size, 766
- font-weight, 766
- fopen(), 79, 80, 82, 83, 89, 90, 150, 436, 521, 556
- for, 73
- FOR NOT FOUND, 306
- foreach, 73, 100, 101, 190
 - tablice, 100
- Format Formularzy Danych, 711
- formatowanie
 - ciągi, 121, 122
 - ciągi do przechowania, 125
 - daty, 46
 - HTML, 122
 - znacznik czasu, 450
- formaty dokumentów, 696
 - PDF, 711
 - PDF, 699
 - RTF, 697
- formaty obrazków, 458
- formaty plików, 85
- Forms Data Format, 711
- formularz zamówienia, 40
- formularze, 40
 - dostęp do zmiennych, 47
 - przetwarzanie, 41
 - wysyłanie plików, 411
 - zmienne, 47
- forum WWW, 671
 - baza danych, 674
 - dodaj_znaki(), 691
 - dodawanie artykułów, 688
 - dyskusja, 672
 - indeks.php, 678
 - konfiguracja forum, 671
 - nowy_artikul.php, 676, 689
 - Phorum, 694
 - pliki aplikacji, 674
 - pobierz_artikul(), 688
 - pobierz_tekst_artikulu(), 691
 - pobierz_tytul_artikulu(), 691
 - projektowanie bazy danych, 674
 - przeglądanie artykułów, 687
 - przeglądanie drzewa artykułów, 676
 - rozwijanie, 678
 - rozwin_wszystko(), 680
 - składniki rozwiązania, 672
 - struktura artykułów, 672
 - struktura systemu, 673
 - stworz_baze.sql, 675
 - wątki, 671
 - wezel_drzewa, 681
 - wezel_drzewa_klasa.php, 681
 - wyswietl_drzewo(), 681
 - wyświetlanie artykułów, 680
 - zachowaj_nowy_artikul(), 692
 - zachowaj_nowy_artikul.php, 692
 - zobacz_artikul.php, 687
 - zwijanie, 678
- fpassstru(), 90
- FPDF, 702
- fputs(), 84
- fread(), 91
- free(), 267
- FreeBSD, 368
- FreeType, 458

- fseek(), 92
- fsockopen(), 522
- ftell(), 92
- FTP, 82, 396, 429, 436
 - anonimowe logowanie, 438
 - kopia bezpieczeństwa pliku, 436
 - kopia lustrzana pliku, 436
 - logowanie się na serwerze, 439
 - mget, 443
 - pobieranie pliku, 441
 - połączenie ze zdalnym serwerem, 439
 - sprawdzanie czasu modyfikacji pliku, 440
 - sprawdzanie wielkości pliku, 443
 - tryby przesyłania plików, 441
 - unikanie przekroczenia dopuszczalnego czasu, 442
 - wysyłanie plików, 442
 - zamykanie połączenia, 442
- ftp://, 82
- FTP_ASCII, 441
- FTP_BINARY, 441
- ftp_connect(), 439
- ftp_fget(), 441, 442
- ftp_fput(), 442
- ftp_get(), 442
- ftp_login(), 439
- ftp_md5(), 440
- ftp_nlist(), 443
- ftp_put(), 442
- ftp_quit(), 440, 442
- ftp_size(), 443
- full join, 245, 249
- FULLTEXT, 232
- func_get_arg(), 155
- func_get_args(), 155
- func_num_args(), 155
- function, 152
- function overloading, 153
- funkcja mieszająca, 338, 378
- funkcje, 141, 149
 - __autoload(), 189
 - __call(), 188, 189
 - __clone(), 188
 - __construct(), 171
 - __destruct(), 171
 - __get(), 170, 180
 - __set(), 170, 180
 - __toString(), 191
 - ==(), 130
- ADDDATE(), 454
- addslashes(), 125, 263, 270, 284, 398, 494, 507
- agregujące, 250
- argumenty, 46
- array(), 98, 100, 609
- array_count_values(), 116
- array_pop(), 111
- array_push(), 111
- array_reverse(), 110, 111, 112
- array_walk(), 115
- arsort(), 107
- asort(), 107
- AVG(), 250
- basename(), 417, 421, 423
- checkdate(), 449
- checkdnsrr(), 436
- chgrp(), 424
- chmod(), 424
- chop(), 121
- chown(), 424
- clearstatcache(), 424
- closedir(), 419
- copy(), 425
- cos(), 723
- count(), 112, 116
- COUNT(), 250
- crypt(), 378
- current(), 114
- date(), 46, 445, 518, 771
- date_add(), 454
- date_diff(), 454
- DATE_FORMAT(), 450, 452
- date_sub(), 454
- DATEDIFF(), 455
- decoct(), 424
- define(), 53
- definiowanie, 152
- deklaracja, 152
- deserialize(), 758
- die(), 495
- dirname(), 421, 423
- disk_free_space(), 421
- dns_get_mx(), 434, 436
- doubleval(), 270
- each(), 114, 115
- empty(), 66, 121, 483
- end(), 114
- ereg_replace(), 139
- eregi_replace(), 139
- error_reporting(), 529
- escapeshellcmd(), 360, 398, 427
- eval(), 494
- exec(), 359, 425
- exit(), 495
- explode(), 113, 127, 128, 130, 436
- extract(), 117, 118
- fclose(), 84, 86, 442
- fdf_create(), 711
- fdf_set_file(), 711
- fdf_set_value(), 711
- feof(), 89, 91
- fgetc(), 91
- fgetcsv(), 89, 90
- fgets(), 89, 553
- fgetss(), 89
- file(), 90, 112
- file_exists(), 92, 440
- file_get_contents(), 84, 91, 432

file_put_contents(), 84
fileatime(), 423
filegroup(), 423, 424
filemtime(), 423, 440
fileopen(), 432
fileowner(), 423, 424
fileperms(), 424
filesize(), 92, 424
filetype(), 424
floatval(), 67
flock(), 93
fopen(), 79, 80, 82, 83, 89, 90, 150, 436, 521, 556
fpassthru(), 90
fputs(), 84
fread(), 91
fseek(), 92
fsockopen(), 522
ftell(), 92
ftp_connect(), 439
ftp_fget(), 441, 442
ftp_fput(), 442
ftp_get(), 442
ftp_login(), 439
ftp_md5(), 440
ftp_nlist(), 443
ftp_put(), 442
ftp_quit(), 440, 442
ftp_size(), 443
func_get_arg(), 155
func_get_args(), 155
func_num_args(), 155
fwrite(), 84, 85
get_current_user(), 497
get_extension_funcs(), 496
get_loaded_extensions(), 496
get_magic_quotes_gpc(), 126, 263, 494
get_magic_quotes_runtime(), 494
getdate(), 448
getenv(), 427, 428
gethostbyaddr(), 435
gethostbyname(), 434, 435
getlastmod(), 497
getmxrr(), 436
gettype(), 65
gregoriantojd(), 456
header(), 463, 471, 481, 709, 723
highlight_file(), 498
highlight_string(), 499
htmlentities(), 355
htmlspecialchars(), 263, 355, 399
imagecolorallocate(), 461, 471
imagecreatefromgif(), 461, 467
imagecreatefromjpeg(), 461, 467
imagecreatefrompng(), 461, 467, 468
imagecreatetruecolor(), 460
imagedestroy(), 464, 471, 478
imagefill(), 462
imagefilledrectangle(), 476, 478
imagejpeg(), 463
imageline(), 462, 476
imagepng(), 463, 468, 478
imagerectangle(), 476, 478
imagestring(), 462
imagesx(), 468
imagesy(), 468
imageftbbox(), 469, 470
imagefttext(), 469, 471
imap_body(), 601, 621, 622
imap_close(), 601, 619, 622
imap_delete(), 601, 623
imap_expunge(), 601, 623
imap_fetchheader(), 601, 621, 622
imap_header(), 601, 621, 622
imap_headers(), 601, 619, 621
imap_open(), 601, 618, 627
implode(), 127, 128
include(), 141, 142, 156, 282, 391
include_once(), 142
ini_get(), 498
ini_set(), 498
intval(), 67, 114
is_array(), 66
is_bool(), 66
is_dir(), 424
is_double(), 66
is_executable(), 424
is_file(), 424
is_float(), 66
is_int(), 66
is_integer(), 66
is_link(), 424
is_long(), 66
is_null(), 66
is_object(), 66
is_readable(), 424
is_real(), 66
is_resource(), 66
is_string(), 66
is_uploaded_file(), 416
is_writable(), 424
isError(), 274
isset(), 66, 67, 159, 483
jdtojulian(), 456
join(), 127
kalendarze, 455
krsort(), 107
ksort(), 107, 109
list(), 432
lstat(), 424
ltrim(), 121, 122
mail(), 121, 430, 554, 600, 601, 631, 668
MailVal(), 140
MAX(), 250
md5(), 378
microtime(), 455
MIN(), 250

funkcje

- mkdir(), 421
- mktime(), 447, 448, 453
- move_uploaded_file(), 416
- mssql_escape_string(), 366
- mysql_error(), 495
- mysqli_affected_rows(), 270
- mysqli_close(), 267
- mysqli_connect(), 264, 507, 521
- mysqli_connect_errno(), 264
- mysqli_errno(), 521
- mysqli_error(), 521
- mysqli_escape_string(), 354
- mysqli_fetch_assoc(), 266
- mysqli_fetch_object(), 267
- mysqli_fetch_row(), 266, 306
- mysqli_free_result(), 267
- mysqli_num_rows(), 266, 270
- mysqli_query(), 265, 306
- mysqli_real_escape_string(), 354, 366, 496
- mysqli_select_db(), 265
- mysqli_stmt_bind_param(), 271
- mysqli_stmt_bind_result(), 271
- mysqli_stmt_execute(), 271
- mysqli_stmt_fetch(), 271
- mysqli_stmt_prepare(), 271
- nazwy, 152, 153
- next(), 114, 115
- nl2br(), 92, 122, 123, 356
- NOW(), 455
- number_format(), 63, 455
- opendir(), 419
- parametry, 46, 150, 154
- parse_url(), 435
- passthru(), 426
- password(), 283
- pdf_add_bookmark(), 715
- pdf_begin_page(), 715
- pdf_close(), 717
- pdf_closepath(), 722
- pdf_continue_text(), 716
- pdf_end_page(), 716
- pdf_fill(), 722, 723
- pdf_findfont(), 716
- pdf_get_buffer(), 717
- pdf_lineto(), 722
- pdf_load_image(), 722
- pdf_moveto(), 722
- pdf_new(), 714, 720
- pdf_open_file(), 714, 717, 720
- pdf_rect(), 721, 722
- pdf_set_info(), 714, 715
- pdf_set_text_pos(), 716
- pdf_setcolor(), 722
- pdf_setlinewidth(), 721, 722
- pdf_show(), 716
- pdf_show_boxed(), 716
- pdf_show_xy(), 722
- pdf_stringwidth(), 722
- pdf_stroke(), 721, 722
- phpinfo(), 53, 149, 428, 600, 702, 800, 807
- popen(), 426
- pos(), 114
- posix_getgrgid(), 424
- posix_getpwuid(), 423, 424
- prev(), 114, 115
- print(), 122, 514
- print_r(), 526
- printf(), 122, 123, 271
- proc_close(), 426
- proc_open(), 426
- prototyp, 150
- przeciążanie, 153
- przekazywanie parametrów, 158
- putenv(), 406, 427, 428, 468
- range(), 99, 111
- readdir(), 419
- readfile(), 90, 91
- rekurencja, 161
- rename(), 425
- require(), 141, 142, 143, 146, 156, 391
- require_once(), 142, 356
- reset(), 114
- return, 160
- rewind(), 92
- rewinddir(), 419
- rmdir(), 421, 422
- rsort(), 107
- rtrim(), 122
- rysowanie, 462
- scandir(), 420
- SEEK_CUR(), 93
- SEEK_END(), 93
- SEEK_SET(), 93
- serialize(), 495, 758
- session_destroy(), 483, 485
- session_get_cookie_params(), 481
- session_set_cookie_params(), 481
- session_start(), 482, 483, 484, 489, 608
- session_unregister(), 483, 485
- session_unset(), 483
- set_error_handler(), 530
- set_magic_quotes_runtime(), 494
- set_time_limit(), 443
- setcookie(), 480
- settype(), 65
- sha1(), 378
- SHA1(), 379
- show_source(), 498
- shuffle(), 110
- simplexml_load_file(), 751
- sin(), 723
- sizeof(), 116
- sort(), 98, 106
- split(), 139
- sprintf(), 122
- stat(), 424

G

- STD(), 250
- STDDEV(), 250
- str_replace(), 132, 709
- strcasecmp(), 129
- strchr(), 131
- strcmp(), 129, 130
- strftime(), 450
- strip_tags(), 151, 399, 416, 481
- stripslashes(), 125, 127, 151, 263, 284, 398, 494
- stristr(), 131
- strlen(), 130
- strnatcmp(), 129
- strpos(), 131, 183
- strrchr(), 131
- strrpos(), 131
- strstr(), 131, 520, 556
- strtok(), 128, 130
- strtolower(), 125
- strtoupper(), 125
- strval(), 67
- substr(), 128, 129
- substr_replace(), 132
- SUM(), 250
- system(), 407, 426
- tempnam(), 406
- time(), 448
- touch(), 425
- trigger_error(), 529
- trim(), 121, 122, 263
- trygonometryczne, 723
- uasort(), 109
- ucfirst(), 125
- ucwords(), 125
- uksort(), 109
- umask(), 422
- UNIX_TIMESTAMP(), 450
- unlink(), 92, 407, 425
- unserialize(), 495, 496
- unset(), 66, 483, 485
- urlencode(), 380, 432
- usort(), 108, 109, 110
- var_export(), 192
- vprintf(), 124
- vsprintf(), 124
- wywołanie, 46, 149
- wywołanie niezdefiniowanej funkcji, 151
- zasięg, 156
- zwracanie wartości, 109, 159, 160
- funkcje dostępne, 170
- funkcje składowane, 303
 - parametry, 303
 - RETURN, 303
 - usuwanie, 304
 - wyświetlanie kodu, 304
 - wywołanie, 304
 - zmienne lokalne, 304
- funkcje_ajax.js, 772
- fwrite(), 84, 85
- gd, 721
- GD, 457
- gd2, 795
- GD2, 457, 468, 478
- GDFONTPATH, 468
- generowanie
 - dokumenty PDF, 713
 - dokumenty w formacie Microsoft Word, 697
 - obrazki, 457
- GET, 609, 765
- get_current_user(), 497
- get_extension_funcs(), 496
- get_loaded_extensions(), 496
- get_magic_quotes_gpc(), 126, 263, 494
- get_magic_quotes_runtime(), 494
- getAllResponseHeaders(), 769
- getStatusCode(), 197
- getdate(), 448
- getElementsByTagName(), 772
- getenv(), 427, 428
- getFile(), 197
- gethostbyaddr(), 435
- gethostbyname(), 434, 435
- getlastmod(), 497
- getLine(), 197
- getMessage(), 197
- getmxrr(), 436
- getProxy(), 756
- getResponseHeader(), 769
- getTrace(), 197
- getTraceAsString(), 197
- gettype(), 65
- Ghostscript, 699
- gid, 424
- GIF, 457, 459
- global, 54, 157
- GMail, 599
- GNU, 29
- Gnu Privacy Guard, 401
- GoDaddy, 339
- Gogle, 729
- GPG, 401, 591
 - deszyfrowanie wiadomości, 404
 - generowanie kluczy, 402
 - importowanie wydzielonego klucza, 403
 - instalacja, 401
 - klucz do podpisywania wiadomości, 403
 - klucz publiczny, 402
 - szyfrowanie wiadomości, 404
 - testowanie, 404
- gpg.exe, 402
- GPL, 29
- GRANT, 221, 224, 225, 275, 281
- Graphics Interchange Format, 459

gregoriantojd(), 456
 GROUP BY, 251
 grupowanie danych, 250
 Gsview, 699
 gunzip, 401

H

haker, 350
 handle, 420
 hasła, 282, 327, 334, 400, 534
 przechowywanie, 376
 szyfrowanie, 378
 ustawianie zapomnianych haseł, 551
 zmiana, 549
 HAVING, 251
 HEAD, 765
 header(), 463, 471, 481, 709, 723
 HEAP, 298
 heredoc, 50
 highlight_file(), 498
 highlight_string(), 499
 host, 276, 278
 Hotmail, 599
 htdocs, 399
 HTML, 28, 40, 696, 697, 726
 HTML entities, 355
 HTML MIME, 430
 htntlentities(), 355
 htmlspecialchars(), 263, 355, 399
 httpasswd, 385
 HTTP, 82, 380, 396, 429, 430, 764
 bezstanowość, 479
 odpowiedzi, 764
 podstawowe uwierzytelnianie, 381
 żądania, 764
 HTTP/1.1, 381, 430
 http://, 82, 556
 HTTP_SERVER_VARS, 80
 HTTP_SESSION_VARS, 485
 httpd, 363
 httpd.conf, 363, 498, 800
 HTTPS, 394, 430, 432, 802
 Hypertext Markup Language, 697
 Hypertext Transfer Protocol, 429, 764

I

IDE, 511, 516
 IDEA, 337
 identyczność, 59
 identyfikacja użytkowników, 349, 373, 534
 identyfikacja właściciela skryptu, 497
 identyfikatory, 51
 MySQL, 232
 sesja, 480, 481
 użytkownik, 424

IETF, 429
 if, 67
 else, 68
 elseif, 69
 IF, 307
 IGNORE, 255, 257
 IINT, 701
 IIS, 339, 340, 361, 363, 793
 konfiguracja, 363
 imagecolorallocate(), 461, 471
 imagecreatefromgif(), 461, 467
 imagecreatefromjpeg(), 461, 467
 imagecreatefrompng(), 461, 467, 468
 imagecreatetruecolor(), 460
 imagedestroy(), 464, 471, 478
 imagefill(), 462
 imagefilledrectangle(), 476, 478
 imagejpeg(), 463
 imageline(), 462, 476
 ImageMagick, 457
 imagepng(), 463, 468, 478
 imagerectangle(), 476, 478
 imagestring(), 462
 imagesx(), 468
 imagesy(), 468
 imagettfbbox(), 469, 470
 imagettftext(), 469, 471
 IMAP, 396, 599, 600, 619
 imap_body(), 601, 621, 622
 imap_close(), 601, 619, 622
 imap_delete(), 601, 623
 imap_expunge(), 601, 623
 imap_fetchheader(), 601, 621, 622
 imap_header(), 601, 621, 622
 imap_headers(), 601, 619, 621
 imap_open(), 601, 618, 627
 IMAP4, 430
 implementacja
 interfejs, 177
 kontrola dostępu, 374
 kontrola wersji, 510
 koszyk na zakupy, 577
 sesja, 482
 system płatności, 564, 589
 uwierzytelnianie, 485, 540
 własna metoda uwierzytelniania, 388
 implode(), 127, 128
 IN, 244, 253, 302, 303
 include(), 141, 142, 156, 282, 391
 include_once(), 142
 include_path, 808
 indeks, 98, 99
 indeksowo-sekwencyjna metoda dostępu, 298
 indeksy, 232, 292
 FULLTEXT, 232
 tworzenie, 232
 INDEX, 223
 Indexed Sequential Access Method, 298

inetpub, 399
informacje

- aktualny katalog, 421
- baza danych, 284
- czas, 445
- data, 445
- data modyfikacji skryptu, 497
- kolumny, 286
- pliki, 422
- środowisko PHP, 496
- założowane rozszerzenia, 496

`ini_get()`, 498
`ini_set()`, 498
inicjowanie tablicy, 100

- tablice indeksowane numerycznie, 98

inkrementacja, 57
INNER JOIN, 245, 249
InnoDB, 295, 298, 564

- transakcje, 300

INOUT, 302
INSERT, 223, 226, 240, 267

- DELAYED, 241
- LOW_PRIORITY, 241
- ON DUPLICATE KEY UPDATE, 241

INSERT INTO, 240
instalacja

- GPG, 401
- OpenSSL, 798
- PEAR, 808
- zapora sieciowa, 368

instalacja Apache, 794

- przy użyciu binariów, 794
- przy użyciu kodów źródłowych, 794
- Unix, 794
- Windows, 802, 804

instalacja MySQL, 794

- Unix, 794, 795
- Windows, 802, 803

instalacja PHP, 793, 794

- Unix, 794, 797
- Windows, 802, 806

`instanceof`, 62, 186
instrukcje

- PHP, 44
- przygotowywane, 270
- SGML, 43
- warunkowe, 67

INT, 234
Integer, 51
INTEGER, 234
integralność plików, 330
Integrated Development Environment, 511
Inteligentny Formularz Pocztowy, 119, 137
interakcja z systemem plików, 411
interakcja ze środowiskiem, 427
interaktywna witryna WWW, 503
interface, 178
interfejs administratora, 565

interfejs użytkownika, 142
interfejsy, 166, 177

- definicja, 178
- implementacja, 177
- Iterator, 190

interfejsy zewnętrzne systemu, 142
International Standard Book Number, 211
Internet Engineering Task Force, 429
Internet Message Access Protocol, 430
Internet Protocol, 396
internetowa baza danych, 207, 259

- architektura, 215
- tworzenie, 217

interpreter skryptów, 215
intl, 33
`intval()`, 67, 114
inżynieria oprogramowania, 504
IP, 396
IS NOT NULL, 244
IS NULL, 244
`is_array()`, 66
`is_bool()`, 66
`is_dir()`, 424
`is_double()`, 66
`is_executable()`, 424
`is_file()`, 424
`is_float()`, 66
`is_int()`, 66
`is_integer()`, 66
`is_link()`, 424
`is_long()`, 66
`is_null()`, 66
`is_object()`, 66
`is_readable()`, 424
`is_real()`, 66
`is_resource()`, 66
`is_string()`, 66
`is_uploaded_file()`, 416
`is_writable()`, 424
ISAM, 298
ISAPI, 802
ISBN, 211, 230
`isError()`, 274
ISO 8601, 450
Isolation, 299
ispell, 553
`isset()`, 66, 67, 159, 483
iteracje, 71, 190
Iterator, 190, 191
iteratory, 190
izolacja, 299

J

Java Server Pages, 30
JavaScript, 43, 44, 373, 616, 767
`jdtojulian()`, 456

- jeden do jednego, 210
 - jeden do wielu, 210
 - język
 - DDL, 240
 - DML, 240
 - JavaScript, 767
 - PHP, 28, 39
 - SGML, 726
 - skryptowy działający po stronie serwera, 28
 - SQL, 29, 239
 - XHTML, 766
 - XML, 725, 726, 768
 - język definiowania danych, 240
 - język manipulowania danymi, 240
 - Język Opisu Usług Sieciowych, 730
 - JFIF, 459
 - join, 245
 - join condition, 246
 - join(), 127
 - Joint Photographic Experts Group, 459
 - JPEG, 458, 459
 - JSP, 30, 768
- K**
- kadr obrazka, 460
 - kalendarze, 455, 456
 - calendar, 456
 - konwersje, 456
 - kampanie marketingowe, 314
 - kapsułkowanie, 166, 170
 - kardynalność tablicy, 117
 - karty kredytowe, 400, 591
 - kaskadowe arkusze stylów, 144, 765, 766
 - kasowanie pliku, 92
 - katalogi, 418
 - dir, 420
 - handle, 420
 - ilość wolnego miejsca, 421
 - informacje, 421
 - odczytywanie danych, 418
 - otwieranie, 419
 - path, 420
 - przeglądanie, 419
 - ścieżka, 420
 - tworzenie, 421
 - usuwanie, 421, 422
 - wyświetlanie zawartości, 418
 - katalogi stron, 433
 - klasy, 166, 168
 - abstrakcyjne, 188
 - API Reflection, 192
 - atrybuty, 166, 168, 170
 - definiowanie, 168, 178
 - destrukторы, 169
 - dir, 420
 - dziedziczenie, 167, 173
 - egzemplarzowanie, 169
 - Exception, 195, 196, 197
 - funkcje dostępowe, 170
 - implementacja interfejsu, 177
 - interfejsy, 177
 - Iterator, 191
 - iteratory, 190
 - konstrukторы, 168
 - kontrola dostępu, 172
 - Mail, 669
 - Mail_mime, 668
 - metody, 183
 - metody statyczne, 186
 - modyfikatory dostępu, 170
 - mysqli, 264
 - nazwy, 179
 - operacje, 166, 183
 - pochodne, 167
 - podklasy, 167
 - przekształcanie w łańcuchy znaków, 191
 - Reflection, 192
 - sprawdzanie typu, 186
 - stałe, 186
 - tworzenie egzemplarzy, 169
 - tworzenie kodu, 179
 - wskazywanie typu, 186
 - wyświetlanie wartości atrybutów, 192
 - wywołanie operacji, 172
 - zakres widoczności, 168
 - zapobieganie dziedziczeniu, 176
 - klasy CSS, 767
 - klasy znaków, 134
 - klasyczna architektura internetowej bazy danych, 215
 - klient
 - poczta elektroniczna, 599
 - SOAP, 726, 756
 - SSH, 439
 - klonowanie obiektów, 188
 - klucz, 98, 99
 - klucz (baza danych), 209, 214
 - obcy, 209, 301
 - podstawowy, 209, 229
 - kod HTML, 28, 40
 - kod łatwy w utrzymaniu, 506
 - dokumentacja, 510
 - dzielenie kodu, 509
 - dzielenie wewnętrznych funkcji, 510
 - komentowanie kodu, 507
 - konwencje nazewnictwa, 506
 - standardy kodowania, 506
 - struktura katalogów, 509
 - wcinanie, 508
 - kod PHP, 28, 42, 499
 - kod źródłowy MySQL, 35
 - kolory, 461
 - kolumny, 208
 - typy danych, 233

- komentarze, 45
 - styl C/C++, 45
 - wielowierszowe, 45
- komentowanie kodu, 507
- komercyjne strony WWW, 311
 - awarie sprzętu komputerowego, 322
 - bezpieczeństwo, 325
 - błędy w oprogramowaniu, 323
 - dostarczanie usług lub wyrobów mających postać cyfrową, 319
 - dostawcy usług, 322
 - konkurencja, 322
 - ograniczanie kosztów, 320
 - ograniczenie pojemności systemów, 323
 - plan działalności, 323
 - przyciągnięcie niewystarczającej liczby klientów, 321
 - przyjmowanie zamówień na produkty i usługi, 315
 - publikowanie informacji w broszurach internetowych, 312
 - ryzyko, 320
 - sprzedaż on-line, 315
 - wyбір strategii, 323
 - zabezpieczenia, 321
 - zagrożenia, 320, 326
 - zmiany polityki rządowej, 323
 - zwiększanie wartości produktów i usług, 319
- komercyjne witryny WWW, 311
- Komodo, 511
- kompresja LZW, 460
- komunikaty o błędach, 61, 151, 518
- komunikaty SOAP, 730
- konfiguracja
 - cookies, 480
 - kontrola sesji, 485
 - obsługa obrazków, 457
 - serwer WWW, 362
- konfiguracja strefy zdemilitaryzowanej, 368
- koniec pliku, 89
- konkatenacja, 46
- konstruktory, 168
- kontrola danych pochodzących od użytkownika, 398
- kontrola dostępu, 374
- kontrola sesji, 479
 - \$_SESSION, 479
 - cookie, 479, 480, 481
 - identyfikator sesji, 481
 - implementacja sesji, 482
 - konfiguracja, 485
 - niszczenie sesji, 483
 - opcje konfiguracyjne sesji, 486
 - przechowywanie identyfikatora sesji, 481
 - SID, 481
 - stosowanie zmiennych sesji, 483
 - URL, 479, 480, 481
 - usuwanie zmiennych, 483
 - uwierzytelnianie, 485
 - zgłaszanie zmiennych sesji, 482
- kontrola wersji, 510
 - CVS, 510
 - wprowadzanie, 510
 - wyprowadzanie, 510
- konwencje nazewnictwa, 506
- konwersja czasu i daty do znacznika czasu Uniksa, 447
- konwersja
 - kalendarz gregoriański na juliański, 456
 - pomiędzy formatami daty PHP i MySQL, 450
 - pomiędzy systemami kalendarzowymi, 455
 - tablice na zmienne skalarne, 117
- kończenie wykonania skryptu, 75, 495
- kopia bezpieczeństwa, 436
- kopia lustrzana pliku, 436
- kopia zapasowa, 329, 342
 - AMANDA, 342
 - MySQL, 292, 342
 - pliki, 342
- kopiowanie plików, 425
- korektor pisowni, 553
- korzeń, 673
- koszty, 333
- koszyk na zakupy, 563, 741, 758
 - baza danych, 568
 - dodawanie produktów, 582
 - FishCartSQL, 598
 - implementacja, 577
 - pliki aplikacji, 567
 - podgląd, 580
 - system płatności, 564
 - śledzenie zakupów użytkownika, 564
- kotwiczenie, 136
- KPHPDevelop, 511
- krotki, 208
- krsort(), 107
- kryptografia, 336
- Książkorama, 563, 566
 - admin.php, 591
 - baza danych, 568
 - dodaj_ksiazke.php, 593
 - dodawanie produktów do koszyka, 582
 - funkcje_admin.php, 596
 - funkcje_ksiazki.php, 582
 - funkcje_wyswietl.php, 580
 - Idź do kasy, 584
 - implementacja bazy danych, 568
 - implementacja interfejsu administratora, 591
 - implementacja katalogu online, 570
 - implementacja koszyka na zakupy, 577
 - implementacja płatności, 589
 - indeks.php, 571
 - interfejs administratora, 565, 591
 - karty kredytowe, 591
 - kasa, 584
 - kasa.php, 584
 - katalog online, 564, 570
 - kategorie, 571
 - koszyk na zakupy, 577

Książkorama

książka_kz.sql, 568
 oblicz_produkty(), 583
 oblicz_wartosc(), 582
 pliki aplikacji, 567
 płatności, 589
 pobierz_kategorie(), 573
 pobierz_nazwe_kategorii(), 575
 pobyt w kasie, 584
 podgląd koszyka, 580
 pokaz_kat.php, 571, 574
 pokaz_kosz.php, 577, 578
 pokaz_ksiazke.php, 576
 przedstawianie szczegółowych danych książki, 575
 przetwarzanie zamówienia, 589
 przetworz.php, 589
 strona początkowa, 570, 571
 system płatności, 564
 szczegółowe dane książki, 575
 śledzenie zakupów użytkownika, 564
 umiesc_zamowienie(), 587
 widok administratora, 566
 widok użytkownika, 565
 wynik_bd_do_tablicy(), 573
 wyswietl_form_ksiazki(), 596
 wyswietl_kategorie(), 574
 wyswietl_koszyk(), 580
 wyświetlanie książek danej kategorii, 574
 wyświetlanie podsumowania w pasku nagłówka, 584
 zakup.php, 585
 zapisywanie uaktualnionego koszyka, 583
 ksort(), 107, 109
 kursory, 305, 306
 kursy akcji, 431

L

LEAVE, 306
 left join, 247
 LEFT JOIN, 247, 249
 libpng, 458
 liczby, 51, 233
 LIKE, 243, 244
 LIMIT, 252, 257
 Linux, 327, 368
 list(), 101, 432
 litery zstępujące, 470
 LOAD DATA FROM MASTER, 294, 295
 LOAD DATA INFILE, 297
 localhost, 522
 LOCK TABLES, 224, 292
 LOCK_EX, 94
 LOCK_NB, 94
 LOCK_SH, 94
 LOCK_UN, 94
 log_errors, 528

logowanie, 545, 608, 641, 643
 serwer FTP, 439
 serwer MySQL, 219
 LONGBLOB, 237
 LONGTEXT, 237
 LOOP, 306
 LOW_PRIORITY, 241, 255, 257
 lstat(), 424
 ltrim(), 121, 122
 LZW, 460

Ł

ładowanie danych z pliku tekstowego do tabeli, 297
 ładowanie tablic z plików, 112
 łańcuch połączenia, 273
 łączy, 672
 łączenie, 245
 ciągi znaków, 49, 127
 tabela z samą sobą, 249
 tabele, 245
 łączone operatory przypisania, 57

M

MAC, 397
 macierz RAID, 329
 magic_quotes_gpc, 126, 398, 494
 magic_quotes_runtime, 398, 494
 magiczne cudzysłowy, 493
 Mail, 669
 factory(), 669
 send(), 669
 Mail Exchange, 436
 mail(), 121, 430, 554, 600, 601, 631, 668
 Mail_Mime, 631, 668, 808
 mailing list manager, 629
 MailVal(), 140
 manipulowanie ciągami znaków, 119
 maszyny zapisu, 298
 ARCHIVE, 298
 CSV, 298
 HEAP, 298
 InnoDB, 298
 ISAM, 298
 MEMORY, 298
 MERGE, 298
 MyISAM, 298
 Math, 63
 MAX(), 250
 max_connections, 264, 367
 MAX_CONNECTIONS_PER_HOUR, 222
 max_execution_time, 497
 MAX_FILE_SIZE, 413
 MAX_QUERIES_PER_HOUR, 222
 MAX_ROWS, 298

- MAX_UPDATES_PER_HOUR, 222
- MaxClients, 264, 363
- MAX-CONCAT, 253
- mbstring, 362
- MD5, 338, 378, 486
- md5(), 378
- MDB2, 272
 - connect(), 273
 - disconnect(), 274
 - getMessage(), 274
 - isError(), 274
 - liczba zwróconych wierszy, 274
 - łańcuch połączenia, 273
 - numRows(), 274
 - pobieranie wierszy, 274
 - połączenie z bazą danych, 273
 - query(), 274
 - zamykanie połączenia, 274
 - zapytania, 274
- MDB2_FETCHMODE_ASSOC, 274
- mechanizmy zorientowane obiektowo, 31
- MEDIUMBLOB, 237
- MEDIUMINT, 234
- MEDIUMTEXT, 237
- MEMORY, 298
- memory_limit, 418
- menedżer list pocztowych, 629
- MERGE, 298
- Message Authentication Code, 397
- Message Digest 5, 378
- method, 41
- metody, 166
 - __autoload(), 189
 - __call(), 189
 - __clone(), 188
 - __construct(), 168
 - __destruct(), 169
 - __toString(), 191
 - abstrakcyjne, 188
 - dostępowe, 170
 - przeciążanie, 188
 - statyczne, 186
 - wywołanie, 173
- metody oddzielania logiki i zawartości, 513
- metody testowania, 332
- metodyki programowania, 32
- Microsoft Excel, 298
- Microsoft Word, 697
- microtime(), 455
- Miernik Bogactwa Billa Gatesa, 432
- migawki, 295
- mikrosekundy, 455
- MIME, 414, 416, 463, 709
- MIN(), 250
- mirror, 436
- mixed, 65
- mkdir(), 421
- mktime(), 447, 448, 453
- MLM, 629
- MLP Piramida, 629
 - \$ _FILES, 664
 - addHTMLImage(), 669
 - akcje, 640
 - architektura skryptu, 635
 - archiwum listy, 652
 - baza danych list i abonentów, 630
 - formularz tworzenia konta, 642
 - funkcje administratora, 656
 - funkcje uwierz.php, 639
 - implementacja funkcji administratora, 656
 - implementacja funkcji użytkownika, 645
 - implementacja logowania, 641
 - indeks.php, 635
 - komponenty rozwiązania, 630
 - konfiguracja bazy danych, 630, 633
 - logowanie, 641, 643
 - loguj(), 644
 - menu administratora, 656
 - niesubskrybuj(), 653
 - obsługa wysyłania wielu plików, 661
 - pliki aplikacji, 633
 - pobierz_archiwum(), 652
 - pobierz_email(), 645
 - pobierz_info_listy(), 651
 - pobierz_niesubskrybowane_listy(), 649
 - podgląd wiadomości, 664
 - przeglądanie archiwum listy, 652
 - przeglądanie informacji na temat listy, 650
 - przeglądanie list, 646
 - rozsyłanie wiadomości, 665
 - sprawdz_logowanie(), 639
 - sprawdz_uzyt_admin(), 639
 - struktura systemu, 631
 - stworz_baze.sql, 633
 - subskrybuj(), 653
 - tworzenie konta, 641
 - tworzenie listy, 657
 - wylogowanie, 656
 - wypisywanie, 653
 - wyslij(), 666
 - wyslij.php, 661
 - wyswietl_elementy(), 647
 - wyswietl_form_hasla(), 655
 - wyswietl_form_konta(), 641
 - wyswietl_form_poczty(), 660
 - wyswietl_informacje(), 650
 - wysyłanie plików, 630
 - wysyłanie wiadomości, 658
 - wysyłanie wiadomości z załącznikami, 631
 - wysyłanie wielu plików, 661
 - zachowaj_konto(), 642
 - zachowaj_liste(), 658
 - zapisywanie się, 653
 - zmiana hasła, 654
 - zmiana konfiguracji konta, 654
 - zmien_haslo(), 655

- MochiKit, 790
- mod_auth, 383
- mod_auth_mysql, 386, 486
 - Auth_MySQL_DB, 388
 - Auth_MySQL_Encryption_Types, 388
 - Auth_MySQL_Password_Field, 388
 - Auth_MySQL_Password_Table, 388
 - Auth_MySQL_Username_Field, 388
 - instalacja modułu, 386
 - plik .htaccess, 387
- mod_ssl, 340, 798
- model DOM, 765, 789
- Model-View-Controller, 32
- moduł szyfrujący, 336
- modułowy schemat nazywania, 507
- modyfikacja nazw plików, 417
- modyfikatory dostępu, 170, 172
 - dziedziczenie, 174
 - private, 172
 - protected, 172
 - public, 172
- monitor MySQL, 218, 219
- monitorowanie bezpieczeństwa, 347
- monitorowanie sprzedaży, 314
- monitorowanie zdarzeń, 340
- Montreal Protocol On Substances That Deplete The Ozone Layer, 343
- move_uploaded_file(), 416
- mssql_escape_string(), 366
- multipart/form-data, 413
- MVC, 32
- MX, 436
- my.cnf, 294
- my.ini, 294
- MyISAM, 298, 301
- myisamchk, 290, 293
- mysql, 218, 219, 797
- MySQL, 29, 217
 - administracja, 275
 - analiza wykonania zapytania, 286
 - analiza wykorzystania kluczy, 290
 - ANALYZE TABLE, 290
 - AUTO_INCREMENT, 228, 229
 - baza danych, 232
 - baza mysql, 275
 - BIGINT, 234
 - BIT, 234
 - BLOB, 234, 236, 237
 - BOOL, 234
 - CALL, 303
 - CASE, 307
 - CHAR, 234, 237
 - CLOSE, 307
 - CREATE INDEX, 232
 - CREATE PROCEDURE, 302
 - CREATE TABLE, 230
 - czas, 234
 - data, 234
 - DATE, 236
 - DATETIME, 236
 - DEC, 235
 - DECIMAL, 235
 - definiowanie przywilejów, 220
 - definiowanie użytkowników, 220
 - deklaracja podprogramu obsługi, 306
 - DELIMITER, 303
 - DESCRIBE, 231, 286
 - DOUBLE, 235
 - DOUBLE PRECISION, 235
 - DROP FUNCTION, 304
 - DROP PROCEDURE, 304
 - duże obiekty binarne, 234
 - ENUM, 235
 - EXPLAIN, 286
 - FETCH, 306
 - FIXED, 235
 - FLOAT, 235
 - FLUSH PRIVILEGES, 281
 - funkcje agregujące, 250
 - funkcje składowane, 303
 - GRANT, 221, 225, 275, 281
 - hasła, 282, 400
 - HAVING, 251
 - identyfikatory, 232, 233
 - IF, 307
 - indeksy, 232, 292
 - indeksy pełnotekstowe, 232
 - informacje o bazie danych, 284
 - informacje o kolumnach, 286
 - informacje o tabeli, 231
 - instalacja, 794, 795, 803
 - INT, 234
 - INTEGER, 234
 - klucze obce, 301
 - kod źródłowy, 35
 - kontrola dostępu, 279
 - kopia zapasowa, 342
 - koszt, 34
 - kursory, 305
 - LEAVE, 306
 - licencja, 29
 - liczby, 233
 - liczby całkowite, 234
 - liczby zmiennoprzecinkowe, 235
 - LOAD DATA INFILE, 297
 - LOCK TABLES, 292
 - logowanie się, 219
 - lokalizacja katalogów i plików z danymi, 232
 - LONGBLOB, 237
 - LONGTEXT, 237
 - LOOP, 306
 - ładowanie danych z pliku tekstowego do tabeli, 297
 - łańcuchy znaków, 234

- maszyny zapisu, 298
- MEDIUMBLOB, 237
- MEDIUMINT, 234
- MEDIUMTEXT, 237
- nadawanie praw, 221
- NOT NULL, 228
- NUMERIC, 235
- ochrona bazy danych, 282
- odbieranie praw, 221
- odzyskiwanie bazy danych, 342
- OPEN, 306
- OPTIMIZE TABLE, 291
- optymalizacja bazy danych, 291
- optymalizacja tabel, 291
- pętle, 305
- podprogramy, 306
- połączenie serwera ze stronami WWW, 283
- poziomy przywilejów, 223
- PRIMARY KEY, 229
- procedury składowane, 302
- przeñośność, 34
- przywileje, 220, 221, 275, 291
- przywileje administratorów, 224
- przywileje specjalne, 225
- przywileje użytkowników, 223, 283
- przywracanie bazy danych, 293
- REAL, 235
- rejestrwanie użytkownika łączącego się z internetu, 226
- rejestrwanie użytkowników, 220, 221
- REPEAT, 306
- replikacja, 293
- REQUIRE, 222
- REVOKE, 221, 224, 225, 281
- SET, 235, 307
- SHOW, 231, 284
- składnia rozszerzona, 251
- SMALLINT, 234
- SSL, 222
- struktury sterujące, 305
- system przywilejów, 221, 275
- systemowa baza danych, 275
- tabele przywilejów, 279
- TEXT, 236, 237
- TIME, 236
- TIMESTAMP, 234, 236
- TINYBLOB, 237
- TINYINT, 234
- TINYTEXT, 237
- transakcje, 299
- tworzenie bazy danych, 220
- tworzenie indeksów, 232
- tworzenie kopii zapasowej, 292
- tworzenie tabel, 227
- typy danych, 233
- typy kolumn, 229
- typy przywilejów, 223
- UNSIGNED, 229
- UNTIL, 306
- uprawnienia, 220
- USE, 227
- ustawianie wartości zmiennych, 307
- usuwanie bazy danych, 258
- usuwanie rekordów z bazy, 257
- usuwanie tabel, 257
- użytkownicy, 220, 276
- użytkownik root, 219
- VARCHAR, 234, 237
- wartości domyślne, 292
- weryfikacja polecenia, 280
- weryfikacja połączenia, 279
- WHILE, 306
- WITH GRANT OPTION, 222
- wsparcie, 35
- wybór bazy danych, 226
- wydajność, 34
- wyrażenia regularne, 244
- wyświetlanie tabel bazy, 231
- YEAR, 236
- zalety, 34
- zamknięcie kursora, 307
- zasada najmniejszego przywileju, 221
- ZEROFILL, 233
- złączenia, 249
- zmiana przywilejów, 281
- zmiana rekordów, 255
- zmiana struktury tabel, 255
- zmiennne lokalne, 304
- MySQL 5, 35
 - nowości, 35
- MySQL 5.1, 35
- MySQL AB, 35
- mysql.columns_priv, 223
- mysql.db, 223
- mysql.host, 223
- mysql.tables_priv, 223
- mysql.user, 223
- mysql_error(), 495
- mysql_install_db, 220
- mysqladmin, 232, 796
 - flush-privileges, 281
 - reload, 281
- mysqlbinlog, 293
- mysqld, 282
- mysqld_safe, 797
- mysqldump, 292
- mysqldhotcopy, 293
- mysqli, 264, 270, 354, 357
 - multi_query(), 354
 - query(), 354
 - real_escape_string(), 354
- mysqli_affected_rows(), 270
- mysqli_close(), 267
- mysqli_connect(), 264, 507, 521
- mysqli_connect_errmo(), 264

mysqli_errno(), 521
 mysqli_error(), 521
 mysqli_escape_string(), 354
 mysqli_fetch_assoc(), 266
 mysqli_fetch_object(), 267
 mysqli_fetch_row(), 266, 306
 mysqli_free_result(), 267
 mysqli_multi_query(), 354
 mysqli_num_rows(), 266, 270
 mysqli_query(), 265, 270, 306
 mysqli_real_escape_string(), 354, 366, 496
 mysqli_select_db(), 265
 mysqli_stmt_bind_param(), 271
 mysqli_stmt_bind_result(), 271
 mysqli_stmt_execute(), 271
 mysqli_stmt_fetch(), 271
 mysqli_stmt_prepare(), 271
 MySQLnd, 33

N

nagłówki HTTP, 463, 765
 NATIONAL, 237
 nawiasy klamrowe, 508
 nawiązywanie połączenia z serwerem MySQL, 264
 nazwy, 51, 162, 506

- domeny, 82
- elementy, 766
- funkcje, 152, 153, 507
- klasy, 179
- komputery, 433
- pliki, 417
- pola formularza, 41
- zmienne, 47, 51

 NET_DNS, 436
 Network News Transfer Protocol, 430, 600
 Network Solutions, 339
 new, 60, 169
 next(), 114, 115
 NFS, 94
 niszczenie sesji, 483
 nl2br(), 92, 122, 123, 356
 NNTP, 430, 600, 627
 NOT EXISTS, 254
 NOT IN, 244
 NOT LIKE, 244
 NOT NULL, 228, 229, 230
 NOW(), 455
 nowe wiersze, 44
 NTFS, 363
 NULL, 52, 125, 230, 240, 248
 number_format(), 63, 455
 NUMERIC, 235
 numery kart kredytowych, 400
 NuSOAP, 725, 732, 734
 nusoap_client, 756

O

obiekty, 166

- \$this, 170
- atrybuty, 166
- destruktory, 169
- funkcje dostępowe, 170
- iteratory, 190
- klonowanie, 188
- konstruktory, 168
- metody dostępowe, 170
- operacje, 166
- serializacja, 495
- tworzenie, 169
- XMLHttpRequest, 768

 Object, 52
 Object Oriented, 165
 obliczanie dat

- MySQL, 454
- PHP, 453

 obliczanie sum w formularzu, 62
 obrazy, 457

- , 464
- czcionki, 462, 465, 468
- czcionki TrueType, 458
- dopasowanie tekstu do przycisku, 468
- figury, 471
- formaty, 458
- funkcje rysujące, 462
- GD2, 457
- GIF, 457, 459
- ImageMagick, 457
- JPEG, 458, 459
- kolory, 461
- konfiguracja kadru, 467
- konfiguracja obsługi, 457
- PNG, 458, 459
- przyciski, 467
- rozmiar tekstu, 469
- rysowanie, 461, 471
- tekst, 462, 465
- tło, 462
- tworzenie, 460
- tworzenie kadru, 460
- umieszczanie tekstu, 461
- WBMP, 459
- wpisywanie tekstu, 471
- wykresy, 471
- wyświetlanie, 463

 obróbka danych, 94
 obsługa błędów, 195, 202, 529, 530

- błędy wywołania, 523

 obsługa IMAP, 600
 obsługa obrazków, 457
 obsługa plików PostScript, 699
 obsługa POP3, 600
 obsługa SOAP, 732, 755

- obsługa wyjątków, 195
 - ochrona bazy danych, 282
 - ochrona plików, 329
 - ochrona sieci komputerowej przed atakami, 341
 - ODBC, 31, 272
 - odczytywanie
 - cały plik, 90
 - dane z adresu URL, 432
 - dane z katalogów, 418
 - dane z pliku, 79, 87
 - dane z pliku wiersz po wierszu, 89
 - poczta elektroniczna, 430, 619
 - pojedynczy znak z pliku, 91
 - rezultaty zapytań, 266
 - zadana długość pliku, 91
 - oddzielanie logiki i zawartości, 513
 - odnajdywanie podciągów, 139
 - odpowiedzi
 - HTTP, 764, 765
 - REST, 731
 - odstępy, 44
 - odwrotne sortowanie zdefiniowane przez użytkownika, 109
 - odwrotny spamming, 330
 - odwrócone apostrofy, 426
 - odzyskiwanie bazy danych MySQL, 342
 - ograniczanie kosztów, 320
 - określanie wielkości pliku, 92
 - ON DUPLICATE KEY UPDATE, 241
 - onreadystatechange, 769
 - OO, 165
 - OPEN, 306
 - Open Database Connectivity, 272
 - Open Database Connectivity Standard, 31
 - Open Source, 29
 - open(), 769
 - opendir(), 419
 - OpenOffice Writer, 697
 - OpenSSL, 33, 340, 798
 - operacje, 166
 - operandy, 55
 - operatory, 55
 - >, 60
 - ALL, 253
 - ANY, 253
 - arytmetyczne, 55
 - BETWEEN, 244
 - bitowe, 60
 - ciągi znaków, 56
 - EXISTS, 254
 - IN, 244, 253
 - instanceof, 62, 186
 - IS NOT NULL, 244
 - IS NULL, 244
 - kolejność, 63
 - konkatenacja, 46
 - LIKE, 243, 244
 - logiczne, 59, 60
 - łączenie ciągów znaków, 56
 - łączone operatory przypisania, 57
 - new, 60, 169
 - NOT IN, 244
 - NOT LIKE, 244
 - pierwszeństwo, 63
 - porównania, 58, 59
 - przypisanie, 51, 55, 56
 - referencja, 58
 - REGEXP, 243, 244
 - SOME, 253
 - tablicowe, 61, 103
 - tłumienie błędów, 61
 - trójkowy, 60
 - wykonania, 61
 - OPTIMIZE TABLE, 291
 - optymalizacja bazy danych, 291
 - projekt, 291
 - tabele, 291
 - optymalizacja kodu, 514
 - MySQL, 514
 - produkty firmy Zend, 514
 - Zend Optimizer, 514
 - or, 60
 - OR, 244
 - or die, 495
 - ORDER BY, 249, 255, 257
 - organizacja kodu źródłowego, 356
 - osadzanie kodu PHP w HTML, 42
 - Osobista Strona Domowa, 29
 - Otwarty Standard Łączności z Bazami Danych, 31
 - otwarty tekst, 335
 - otwieranie
 - katalogi, 419
 - pliki, 79
 - pliki przez protokół FTP, 82
 - pliki przez protokół HTTP, 82
 - OUT, 302, 303, 307
- ## P
- papier, 696
 - parametry, 46, 150, 154
 - funkcje składowane, 303
 - IN, 302
 - INOUT, 302
 - OUT, 302
 - procedury składowane, 303
 - przekazywanie, 158
 - parent, 176
 - parse_url(), 435
 - parser HTML, 728
 - passthru(), 426
 - password(), 283
 - path, 420
 - PayPal, 729

- PCRE, 33, 133
- PDF, 695, 696, 699
 - pdf_add_bookmark(), 715
 - pdf_begin_page(), 715
 - pdf_close(), 717
 - pdf_closepath(), 722
 - pdf_continue_text(), 716
 - pdf_end_page(), 716
 - pdf_fill(), 722, 723
 - pdf_findfont(), 716
 - pdf_get_buffer(), 717
 - pdf_lineto(), 722
 - pdf_load_image(), 722
 - pdf_moveto(), 722
 - pdf_new(), 714, 720
 - pdf_open_file(), 714, 717, 720
 - pdf_rect(), 721, 722
 - pdf_set_info(), 714, 715
 - pdf_set_text_pos(), 716
 - pdf_setcolor(), 722
 - pdf_setlinewidth(), 721, 722
 - pdf_show(), 716
 - pdf_show_boxed(), 716
 - pdf_show_xy(), 722
 - pdf_stringwidth(), 722
 - pdf_stroke(), 721, 722
- PDFlib, 695, 702, 711, 713, 717
 - hiperłącza, 715
 - inicjacja dokumentu PDF, 714
 - koordynaty, 715
 - liczba stron, 715
 - rozmiar strony, 715
 - rysowanie, 721
 - tworzenie stron, 715
 - zakładki, 715
- PDFlib GmbH, 713
- PDFlib Lite, 702
- PDO, 31
- PEAR, 808
 - pear install, 808
 - pear list, 808
 - pear list-all, 808
 - pear upgrade, 808
- PEAR::MDB2, 272
- PEAR::Net_DNS, 436
- PECL, 32, 457, 806
- perforce, 511
- Perl, 30, 133, 768
- Personal Home Page, 29
- Personal Web Server, 793
- personalizacja użytkowników, 533, 534
- pętle, 71
 - break, 75
 - continue, 75
 - do..while, 74
 - for, 73
 - foreach, 73, 100, 190
 - MySQL, 305
 - przeskok do następnej iteracji, 75
 - tablice, 100
 - while, 72
 - zatrzymanie wykonywania, 75
- PGP, 401, 591
- phar, 33
- Phorum, 694
- php, 499
- PHP, 28, 29, 39, 768
 - \$this, 170
 - abstract, 188
 - alternatywna składnia struktur sterujących, 75
 - API Reflection, 192
 - aplikacje, 40
 - biblioteki, 31
 - blok kodu, 68
 - break, 70, 75
 - calendar, 456
 - catch, 195
 - ciągi znaków, 50
 - class, 168
 - continue, 75
 - czas, 445
 - data, 445
 - declare, 76
 - destruktory, 169
 - do..while, 74
 - dokumentacja, 32
 - dostęp do zmiennych formularza, 47
 - dyrektywy wykonania, 76
 - dziedziczenie, 173
 - else, 68
 - elseif, 69
 - extends, 173
 - final, 176
 - for, 73
 - foreach, 73
 - funkcje, 141, 149
 - funkcje dostępowe, 170
 - GD, 457
 - GD2, 457
 - generowanie obrazków, 457
 - global, 157
 - identyfikacja właściciela skryptu, 497
 - identyfikatory, 51
 - if, 67
 - informacje o środowisku, 496
 - instalacja, 794, 806
 - instanceof, 186
 - instrukcje, 44
 - instrukcje warunkowe, 67
 - integracja z bazami danych, 31
 - interface, 178
 - interfejsy, 177
 - interfejsy bazodanowe, 272
 - iteracje, 71, 190

- iteratory, 190
- klasy, 166, 168
- klasy abstrakcyjne, 188
- klonowanie obiektów, 188
- kod źródłowy, 32
- komentarze, 45
- konfiguracja obsługi obrazków, 457
- kontrola sesji, 479
- kończenie wykonania skryptu, 75, 495
- koszt, 31
- łączenie ciągów znaków, 49
- mechanizmy obiektowe, 31, 165, 186
- metody statyczne, 186
- modyfikatory dostępu, 170, 172
- MySQL, 259
- nazwy, 51
- new, 169
- obiekty, 166
- obsługa błędów, 195
- obsługa wyjątków, 195
- odczytywanie rezultatów zapytań, 266
- odstępy, 44
- operatory, 55
- osadzanie kodu w HTML, 42
- parent, 176
- php.ini, 362
- pierwszeństwo operatorów, 64
- pliki, 79
- podjmowanie decyzji, 67
- podświetlanie źródeł, 498
- połączenie z bazą MySQL, 259
- ponowne wykorzystanie kodu, 141
- powtarzanie działań, 71
- późne wiązania statyczne, 187
- private, 172
- programowanie obiektowe, 165
- protected, 172
- protokoły, 429
- przeciążanie metod, 188
- przenośność, 31
- przerwanie wykonywania fragmentu kodu, 75
- przestrzenie nazw, 162
- przetwarzanie formularza, 41
- public, 172
- return, 159
- rozszerzenia, 496
- rzutowanie typu, 52
- serializacja, 495
- sesje, 479
- sieć, 429
- skalowalność, 30
- stałe, 53
- stałe klasowe, 186
- static, 186
- struktury kontrolujące, 67
- switch, 69
- system plików, 358
- szyfrowanie danych, 401
- tablice, 97
- tablice superglobalne, 48
- throw, 195
- try, 195
- tworzenie obiektów, 169
- typy danych, 51
- usuwanie błędów, 517
- while, 72
- wiersz poleceń, 499
- wsparcie, 32
- wstawianie danych do bazy, 267
- wydajność, 30
- wyjątki, 195
- wykonywanie ciągów, 494
- wysyłanie plików, 411
- wysyłanie zapytań do bazy danych, 265
- wywołanie funkcji, 46
- zalety, 30
- zasięg zmiennych, 54
- zastosowanie, 40
- zmiana środowiska wykonawczego, 497
- zmiennne, 47, 50, 65
- zmiennne zmiennych, 53
- znaczniki, 43
- PHP Database Objects, 31
- PHP Extension Class Library, 457
- PHP Extension Community Library, 702
- PHP Hypertext Preprocessor, 29
- PHP Preprocessor Hipertekstu, 29
- php.ini, 357, 362, 798, 806
- php.ini-dist, 798, 806
- php.ini-recommended, 798, 806
- php_gd2.dll, 458
- PHP_INI_ALL, 498
- PHP_INI_PERDIR, 498
- PHP_INI_SYSTEM, 498
- PHP_INI_USER, 498
- PHP_SELF, 183
- PHP4.0, 151
- PHP5.0, 31, 32
- nowości, 32
- PHP5.3, 33
- PHP6, 33
- phpautodoc, 512
- phpdoc, 512
- PHPDocumentor, 512
- PHPEd, 511
- phpinfo(), 53, 149, 428, 600, 702, 800, 807
- phpMyAdmin, 218
- PHPSESSID, 486, 526
- pierwszeństwo operatorów, 63, 64
- plain text, 335
- PlainText, 388
- planowanie działań na wypadek awarii, 371
- planowanie projektu WWW, 504
- plany działalności, 323

- pliki, 79, 358
 - .htaccess, 149, 383, 498
 - .htpass, 384
 - ASCII, 696
 - atrybuty, 424
 - blokowanie, 93
 - CSV, 90
 - czas ostatniego otwarcia, 423
 - FDF, 711
 - formaty, 85
 - gid, 424
 - GIF, 459
 - identyfikator grupy, 424
 - identyfikator użytkownika, 424
 - informacje, 422
 - jednorodne, 77, 95
 - JPEG, 458, 459
 - kasowanie, 92
 - koniec, 89
 - kopia bezpieczeństwa, 436
 - kopia lustrzana, 436
 - kopiowanie, 425
 - odczytywanie całego pliku, 90
 - odczytywanie danych, 79, 87
 - odczytywanie danych wiersz po wierszu, 89
 - odczytywanie pojedynczego znaku, 91
 - odczytywanie zadanej długości, 91
 - określanie wielkości, 92
 - otwieranie, 79
 - PDF, 695
 - PNG, 458, 459
 - poruszanie się wewnątrz pliku, 92
 - prn, 701
 - problemy z otwieraniem, 82
 - przenoszenie, 425
 - ps, 701
 - rozszerzenia, 143
 - RTF, 697
 - sprawdzanie czasu modyfikacji, 440
 - sprawdzanie istnienia, 92
 - ścieżka dostępu, 80
 - tar.bz2, 401
 - tar.gz, 401
 - tryb otwarcia, 79, 81
 - ttf, 468
 - tworzenie, 425
 - uid, 424
 - uprawnienia, 424
 - ustawianie wskaźnika, 92
 - usuwanie, 425
 - WBMP, 459
 - wielkość, 92, 424
 - zamykanie, 86
 - zapisywanie danych, 79, 84
 - zmiana grupy, 424
 - zmiana uprawnień, 424
 - zmiana właściciela, 424
 - zmiana właściwości, 424
 - znacznik czasu, 423
- PNG, 458, 459
- poczta elektroniczna, 121, 430
 - HTML MIME, 430
 - IMAP4, 430
 - odczytywanie, 430
 - POP, 430
 - protokoły, 599
 - wysyłanie, 430
- podatki, 62
- podjmowanie decyzji, 67
- podklasy, 167
- podpis cyfrowy, 327, 338
- podpisywanie certyfikatu, 339
- podprogramy, 306
- podstawowe uwierzytelnianie, 380
 - htpasswd, 385
 - PHP, 381
 - pliki .htaccess, 383
 - serwer Apache, 383
- podświetlanie źródła, 498
- podwójne sprawdzanie oczekiwanych wartości, 351
- podzapytania, 252
 - EXISTS, 254
 - operatory, 253
 - skorelowane, 254
 - tabele tymczasowe, 254
 - wierszowe, 254
- pola, 208
- pola przywilejów, 278
- pola zakresu, 278
- polecenia rcp, 327
- polimorfizm, 167
- polityka bezpieczeństwa, 333
- polityka prywatności, 317
- połączenie serwera MySQL ze stronami WWW, 283
- połączenie sieciowe, 433
- połączenie SSL, 408
- połączenie z bazą danych, 273
- połączenie z bazą MySQL, 259, 264
 - limit liczby połączeń, 264
 - mysqli, 264, 270
 - mysqli_connect(), 264
 - zamykanie połączenia, 267
- połączenie z serwisami sieciowymi, 522
- połączenie ze zdalnym serwerem FTP, 439
- ponowne wykorzystanie kodu, 141, 505
 - koszt, 142
 - niezawodność, 142
 - spójność, 142
- POP, 396, 430
- POP3, 599
- popen(), 426
- porównania, 58, 59
 - ciągi znaków, 129

- Portable Document Format, 695, 699
- Portable Network Graphics, 459
- poruszanie się wewnątrz pliku, 92
- poruszanie się wewnątrz tablicy, 114
- porządkowanie ciągów, 129
- porządkowanie tablic, 107
- pos(), 114
- POSIX, 133
- posix_getgrgid(), 424
- posix_getpwuid(), 423, 424
- POST, 609, 765
- Post Office Protocol, 430
- post_max_size, 412
- postdekrementacja, 57
- postinkrementacja, 57
- PostScript, 696, 698
- PostScript Type 1, 458
- potok IMAP, 619
- potomek, 167, 673
- potrójny DES, 337
- powrót karetki, 44
- powtarzanie działań, 71
- poziomy zgłaszania błędów, 527
- późne wiązania statyczne, 187
- predekrementacja, 57
- preinkrementacja, 57
- prepare(), 271
- Pretty Good Privacy, 401
- prev(), 114, 115
- prezentacja stron, 766
- PRIMARY KEY, 229, 230
- print, 65
- print(), 122, 514
- print_r(), 526
- printf(), 122, 123, 271
 - typy kodów wyznaczników konwersji, 124
- privacy policy, 317
- private, 168, 172, 174
- problemy z otwieraniem plików, 82
- proc_close(), 426
- proc_open(), 426
- procedury składowane, 302
 - CASE, 307
 - CLOSE, 307
 - deklaracja, 302
 - FETCH, 306
 - IF, 307
 - IN, 302, 303
 - INOUT, 302
 - kursory, 305, 306
 - OUT, 303
 - parametry, 302, 303
 - pętle, 305
 - struktury sterujące, 305
 - treść, 303
 - usuwanie, 304
 - wyświetlanie kodu, 304
 - wywołanie, 303
 - zamknięcie kursora, 307
 - zmienne lokalne, 304
 - znak zakończenia wiersza, 303
- PROCESS, 224, 283
- procesy, 426
- procs_priv, 276, 278, 281
- produkt kartezjański, 245
- programowanie obiektowe, 165
- projekt WWW, 166, 504
 - Bitkeeper, 511
 - kontrola wersji, 510
 - oddzielanie logiki i zawartości, 513
 - optymalizacja kodu, 514
 - perforce, 511
 - planowanie, 504
 - ponowne stosowanie kodu, 505
 - prototypowanie, 512
 - prowadzenie, 504
 - struktura katalogów, 509
 - testowanie, 515
 - wersje beta, 515
- projektowanie bazy danych, 674
 - internetowa baza danych, 207, 211
- Prosty Protokół Udostępniania Obiektów, 726
- Prośby o Komentarz, 429
- protected, 168, 172, 174
- protokoły, 395, 429
 - bezzastanowe, 479
 - FTP, 82, 429, 436
 - HTTP, 82, 380, 396, 429, 764
 - HTTPS, 432, 802
 - IMAP, 599
 - IMAP4, 430
 - NNTTP, 430, 600
 - POP, 430
 - POP3, 599
 - SMTP, 430, 600
 - SOAP, 725, 726, 729, 755
 - SSL, 284, 317, 328, 333, 340, 395
 - TCP/IP, 328, 368, 396
 - TLS, 398
 - warstwa aplikacji, 396
- prototyp funkcji, 150
- Prototype JavaScript Framework, 789
- prototypowanie, 512
 - problemy, 513
- prowadzenie projektu WWW, 504
- przebieg transakcji, 392
- przechowywanie
 - dane, 77, 78, 399
 - hasła dostępu, 376
 - identyfikator sesji, 481
 - zakładki, 535
- przeciążanie funkcji, 153, 169, 188, 520
- przeciwdziałanie atakom DoS, 330
- przeglądanie katalogów, 419
- przeglądy kodu, 515

przekazywanie parametrów, 158
 przez referencję, 158
 przez wartość, 158
 przekształcanie klas w łańcuchy znaków, 191
 przenoszenie plików, 425
 przenośność kodu, 31
 Przenośny Format Dokumentu, 695
 przerwanie wykonywania fragmentu kodu, 75
 przestrzenie nazw, 33, 162
 XML, 728
 przesunięcie bitowe, 60
 przesyłanie
 dane HTTP, 396
 hipertekst, 429
 pliki, 429, 436
 przetwarzanie
 formularze, 41
 pliki, 79
 tablice, 100
 przodek, 167, 673
 przyjmowanie zamówień na produkty i usługi, 315
 przypisanie, 51, 55, 56
 operatory łączone, 57
 zwracane wartości, 56
 przywileje, 221, 275, 291
 przywileje użytkowników, 283
 przywracanie bazy danych MySQL, 293
 ps2pdf, 701
 public, 168, 172, 174
 public_html, 82, 83
 publikowanie informacji w broszurach internetowych,
 312
 putenv(), 406, 427, 428, 468
 PWS, 793

Q

query(), 265, 270, 354
 QUICK, 257

R

RAID, 329, 342
 ramka ograniczająca, 469
 range(), 99, 111
 RC2, 337
 RC4, 337
 RC5, 337
 rcv, 327
 RDBMS, 29, 94, 95, 207, 239
 read(), 420
 readdir(), 419
 readfile(), 90, 91
 readyState, 769, 772
 REAL, 235
 recovery planning, 371
 Redundant Array of Inexpensive Disks, 329

REFERENCES, 224
 referencja, 58
 Reflection, 33, 192
 REGEXP, 243, 244
 register_globals, 47, 48, 362, 414
 register_long_arrays, 47, 362
 reinterpretacja zmiennych, 67
 rejestracja użytkowników, 221, 540
 reklama internetowa, 315
 rekomendacje, 559
 rekomendowanie zakładek, 535
 rekord MX, 436
 rekordy, 208
 rekurencja, 161
 relacje, 210
 jeden do jednego, 210
 jeden do wielu, 210
 wiele do wielu, 210
 relacyjna baza danych, 94, 208
 atrybuty, 208
 klucze, 209
 kolumny, 208
 krotki, 208
 pola, 208
 rekordy, 208
 relacje, 210
 tabele, 208
 wartości, 208
 relational database management system, 29
 RELOAD, 224, 283
 rename(), 425
 REPEAT, 306
 REPLICATION CLIENT, 224
 REPLICATION SLAVE, 224
 replikacja, 293
 FLUSH TABLES WITH READ LOCK, 295
 konfiguracja odbiorcy, 295
 konfiguracja serwera nadrzędnego, 294
 LOAD DATA FROM MASTER, 294, 295
 odbiorcy, 293
 początkowy transfer danych, 294
 przywileje, 294
 serwer nadrzędny, 293
 serwer odbiorcy, 295
 SHOW MASTER STATUS, 295
 transfer danych początkowych, 294
 UNLOCK TABLES, 295
 Representational State Transfer, 726
 Requests for Comments, 121, 429
 REQUIRE, 222
 require(), 141, 142, 143, 146, 156, 391
 rozszerzenia plików, 143
 szablony stron WWW, 144
 require_once(), 142, 356
 reset(), 114
 resource, 52
 responseText, 769
 responseXML, 769

- REST, 725, 726, 750
 - return, 109, 159, 160
 - RETURN, 303
 - RETURNS, 303
 - reverse spamming, 330
 - REVOKE, 221, 224, 225, 281
 - rewind(), 92, 420
 - rewinddir(), 419
 - RFC, 121, 429
 - RFC2616, 430
 - RFC822, 121, 430
 - RGB, 461
 - Rich Text Format, 695, 697
 - rmdir(), 421, 422
 - ROLLBACK, 298, 300
 - root, 219
 - rozczarowani pracownicy, 350
 - rozdzielanie ciągów, 127
 - wyrażenia regularne, 139
 - rozmiar tekstu, 469
 - rozpoczynanie sesji, 482
 - Rozszerzalny Język Oznaczeń, 43, 726
 - rozszerzenia PHP, 496
 - rozszerzenia plików, 143
 - rozwidlanie zewnętrznych procesów, 426
 - równość, 58, 59
 - RSA, 336
 - rsort(), 107
 - RTF, 695, 696, 697
 - rtrim(), 122
 - Ruby, 30
 - rysowanie, 461
 - figury, 471
 - wykresy, 471
 - rzutowanie typu, 52
- S**
- safe_mode_allowed_env_vars, 428
 - SAMBA, 342
 - sans-serif, 766
 - SAPI, 395
 - scandir(), 420
 - schemat bazy danych, 209
 - schemat XML, 728
 - Secure Hash Algorithm 1, 378
 - Secure Shell, 439
 - Secure Sockets Layer, 284, 317
 - SEEK_CUR(), 93
 - SEEK_END(), 93
 - SEEK_SET(), 93
 - SELECT, 223, 226, 242, 267
 - agregowanie danych, 250
 - aliasy, 248
 - ALIKE, 244
 - AS, 248
 - CROSS JOIN, 245
 - GROUP BY, 251
 - grupowanie danych, 250
 - HAVING, 251
 - INNER JOIN, 245
 - LEFT JOIN, 247
 - LIKE, 244
 - LIMIT, 252
 - łączenie tabel, 245
 - łączenie tabeli z samą sobą, 249
 - ograniczanie liczby wierszy, 252
 - ORDER BY, 249
 - podzapytania, 252
 - podzapytania skorelowane, 254
 - REGEXP, 243
 - szeregowanie danych, 249
 - USING, 248
 - WHERE, 243
 - wyszukiwanie danych niepołączonych ze sobą, 247
 - wyszukiwanie danych spełniających określone kryteria, 243
 - wyszukiwanie danych w wielu tabelach, 245
 - złączenia, 245
 - select_db(), 265
 - selektory, 766
 - send(), 769
 - serializacja, 495, 758
 - serialize(), 495, 758
 - Server Side Include, 142
 - SERVER_PORT, 407
 - serwer, 411
 - serwer bazy danych, 215
 - MySQL, 217, 218
 - serwer WWW, 215, 339, 793
 - Apache, 363
 - IIS, 363
 - konfiguracja, 362
 - serwis poczty elektronicznej oparty na WWW, 599
 - IMAP, 600
 - protokoły poczty, 599
 - Świeża poczta, 602
 - sesje, 380, 479, 483, 548, 609
 - cookie, 481
 - identyfikator, 480
 - implementacja, 482
 - niszczenie, 483
 - opcje konfiguracyjne, 486
 - przechowywanie identyfikatora, 481
 - rozpoczynanie, 482
 - stosowanie zmiennych, 483
 - usuwanie zmiennych, 483
 - zmiennne, 480
 - session.auto_start, 482, 486
 - session.cache_expire, 486
 - session.cookie_domain, 486
 - session.cookie_lifetime, 486
 - session.cookie_path, 486
 - session.cookie_secure, 486
 - session.gc_maxlifetime, 482

- session.hash_function, 486
- session.name, 486
- session.save_handler, 486
- session.save_path, 486
- session.use_cookies, 486
- session.use_trans_sid, 481
- session_destroy(), 483, 485
- session_get_cookie_params(), 481
- session_set_cookie_params(), 481
- session_start(), 482, 483, 484, 489, 608
- session_unregister(), 483, 485
- session_unset(), 483
- SET, 235, 237, 307
- set_error_handler(), 530
- set_magic_quotes_runtime(), 494
- set_time_limit(), 443
- Set-Cookie, 480
- setcookie(), 480
- setRequestHeader(), 769
- settype(), 65
- SGML, 43, 726
- SHA, 338
- SHA-1, 378
- sha1(), 378
- SHA1(), 379
- SHOW, 231, 284, 285
- SHOW COLUMNS, 284
- SHOW COLUMNS FROM, 285
- SHOW CREATE DATABASE, 286
- SHOW CREATE FUNCTION, 304
- SHOW CREATE PROCEDURE, 304
- SHOW CREATE TABLE, 286
- SHOW DATABASES, 224, 231, 284, 285
- SHOW ENGINES, 286
- SHOW ERRORS, 286
- SHOW GRANTS, 285, 286
- SHOW INDEX FROM, 285
- SHOW INNODB STATUS, 286
- SHOW MASTER STATUS, 295
- SHOW PRIVILEGES, 286
- SHOW PROCESSLIST, 285
- SHOW STATUS, 285
- SHOW STORAGE ENGINES, 286
- SHOW TABLE STATUS, 286
- SHOW TABLES, 284, 285
- SHOW VARIABLES, 285
- SHOW WARNINGS, 286
- show_source(), 498
- S-HTTP, 394
- shuffle(), 110
- SHUTDOWN, 224, 283
- SID, 481
- sieć, 367, 429
 - ataki DoS, 369
 - połączenia, 433
 - strefa zdemilitaryzowana, 368
 - zapora sieciowa, 368
- Simple Mail Transfer Protocol, 430
- Simple Object Access Protocol, 726
- SimpleXML, 32, 725
- simplexml_load_file(), 751
- sin(), 723
- sizeof(), 116
- składanie zamówienia, 318
- składnia, 517
- składnia heredoc, 50
- składniowy cukier, 505
- skrypty
 - działające po stronie klienta, 767
 - działające po stronie serwera, 768
 - JavaScript, 764, 765
 - PHP, 41
- słowa kluczowe
 - abstract, 188
 - break, 75
 - case, 69
 - catch, 195
 - class, 168, 179
 - clone, 188
 - const, 186
 - continue, 75
 - echo, 65
 - else, 68
 - elseif, 69
 - exit, 75
 - extends, 173
 - final, 176
 - for, 73
 - foreach, 73, 100
 - function, 152
 - global, 54, 157
 - if, 67
 - instanceof, 186
 - interface, 178
 - new, 169
 - parent, 176
 - print, 65
 - private, 168, 172, 174
 - protected, 168, 172, 174
 - public, 168, 172, 174
 - return, 109, 159, 160
 - static, 186
 - switch, 69
 - throw, 195, 197
 - try, 195
 - while, 72, 101
- słownik, 553
- SMALLINT, 234
- Smarty, 514
- SMTP, 396, 430, 600, 601
- SOAP, 725, 726, 729, 755
 - klient, 756
 - obsługa, 755
 - żądanie, 729

- SOME, 253
- sort(), 98, 106
- SORT_NUMERIC, 107
- SORT_REGULAR, 107
- SORT_STRING, 107
- sortowanie odwrotne, 107
- sortowanie tablic, 106
- sortowanie tablic wielowymiarowych, 108
 - typy sortowań definiowane przez użytkownika, 108
- SourceForge, 512
- spacje, 44
- Specjalna Grupa Inżynierii Internetu, 429
- spersonalizowane dokumenty PDF, 695
- SPL, 33
- split(), 139
- spójność, 299
- sprawdzanie
 - czas modyfikacji pliku, 440
 - długość ciągu, 130
 - istnienie pliku, 92
 - obecność kodu SQL w ciągach znaków, 354
 - oczekiwane wartości, 351
 - poprawność adresu pocztowego, 138, 140
 - poprawność daty, 449
 - poprawność wpisanych danych, 263
 - stan zmiennej, 66
 - typ klasy, 186
 - typ zmiennej, 65
- sprintf(), 122
- sprzedaż on-line, 315
 - dane osobiste, 317
 - polityka prywatności, 317
 - prostota obsługi, 318
 - pytania bez odpowiedzi, 316
 - składanie zamówienia, 318
 - towary, 316
 - wpływ na złożenie zamówienia
 - przez potencjalnego klienta, 316
 - zaufanie, 317
 - zgodność, 318
- sqlite3, 33
- SQL, 29, 239
 - agregowanie danych, 250
 - aliasy, 248
 - ALL, 253
 - ALTER TABLE, 255
 - ANY, 253
 - CREATE TABLE, 227
 - CROSS JOIN, 245
 - DELETE, 257
 - DESCRIBE, 286
 - DROP DATABASE, 258
 - DROP TABLE, 258
 - EXISTS, 254
 - GROUP BY, 251
 - grupowanie danych, 250
 - HAVING, 251
 - IN, 253
 - INNER JOIN, 245
 - INSERT, 240
 - ORDER BY, 249
 - podzapytania, 252
 - SELECT, 242
 - SOME, 253
 - szeregowanie danych, 249
 - UPDATE, 255
 - usuwanie bazy danych, 258
 - usuwanie rekordów z bazy, 257
 - usuwanie tabel, 257
 - WHERE, 243
 - wyszukiwanie danych, 242
 - wyszukiwanie danych spełniających określone kryteria, 243
 - wyszukiwanie danych w wielu tabelach, 245
 - zapisywanie danych do bazy, 240
 - złączenia, 245
 - zmiana rekordów w bazie danych, 255
 - zmiana struktury tabel, 255
- SQLiteException, 306
- SQLite Embeddable SQL Database Engine, 33
- SQLWARNING, 306
- SSH, 439
- SSL, 222, 284, 317, 328, 333, 340, 394, 584, 800
 - kod uwierzytelniający, 397
 - MAC, 397
 - nagłówki TCP, 397
 - operacja uzgadniania, 396
 - serwer WWW, 340
 - wysyłanie danych, 397
 - zabezpieczanie transmisji danych, 396
- SSL 3.0, 398
- stabilność kodu, 358
- stałe, 53
- stałe klasowe, 186
- stan zmiennej, 66
- Standard Generalized Markup Language, 43, 726
- Standardowy Ogólny Język Oznaczenia, 43
- standardy internetowe, 429
- standardy kodowania, 506
- START TRANSACTION, 300
- stat(), 424
- static, 186
- status, 769
- statusText, 769
- STD(), 250
- STDDEV(), 250
- sterowniki ODBC, 31
- stos protokołu, 396
- stos wywołań, 197
- stosowanie zmiennych sesji, 483
- str_replace(), 132, 709
- strcasecmp(), 129
- strchr(), 131
- strcmp(), 129, 130

- strefa zdemilitaryzowana, 350, 368
 - strftime(), 450
 - kody formatowania, 451
 - String, 51
 - strip_tags(), 151, 399, 416, 481
 - stripslashes(), 125, 127, 151, 263, 284, 398, 494
 - stristr(), 131
 - strlen(), 130
 - strnatcmp(), 129
 - strona WWW, 311, 431
 - wykonywanie zapytań, 265
 - Stronghold, 340
 - strpos(), 131, 183
 - strrchr(), 131
 - strrpos(), 131
 - strstr(), 131, 520, 556
 - strtok(), 128, 130
 - strtolower(), 125
 - strtoupper(), 125
 - Structured Query Language, 29, 239
 - struktura drzewiasta, 672
 - struktura katalogów, 509
 - Strukturalny Język Zapytań, 29
 - struktury kontrolujące, 67
 - strval(), 67
 - style CSS, 513
 - style znaczników PHP, 43
 - su, 340
 - substr(), 128, 129
 - substr_replace(), 132
 - SUM(), 250
 - Summary, 314
 - sumy, 62
 - SUPER, 224
 - superglobalne tablice, 48
 - switch, 69
 - break, 70
 - case, 69
 - default, 71
 - system kontroli wersji, 510
 - system MLP, 629
 - system oceniania online, 700
 - aplikacje tworzące PDF, 702
 - generowanie dokumentów PDF, 713
 - indeks.html, 704
 - nagłówki, 723
 - ocena odpowiedzi, 705
 - oprogramowanie generujące dokumenty, 700
 - pdf.php, 712
 - PDFlib, 717
 - pdflib.php, 717
 - pliki aplikacji, 703
 - pytania quizowe, 704
 - rtf.php, 709
 - składniki rozwiązania, 700
 - szablony PDF, 700, 711
 - szablony RTF, 700
 - tworzenie certyfikatu, 717
 - tworzenie certyfikatu PDF z szablonu, 710
 - tworzenie certyfikatu RTF, 707
 - wynik.php, 705
 - zadawanie pytań, 704
 - system operacyjny, 369
 - system plików, 94, 358, 411, 422
 - informacje o pliku, 422
 - zmiana właściwości pliku, 424
 - system płatności, 564, 589
 - system przywilejów, 275
 - MySQL, 221
 - system szablonów, 514
 - System Wersji Współbieżnych, 510
 - system zakładek online, 533
 - system zarządzania relacyjnymi bazami danych, 29, 94
 - system(), 407, 426
 - systemowa baza danych, 275
 - columns_priv, 278, 281
 - db, 278, 279
 - host, 278, 280
 - poła przywilejów, 278
 - poła zakresu, 278
 - procs_priv, 278, 281
 - tables_priv, 278, 280
 - user, 276, 277
 - szablony, 514
 - szablony stron WWW, 144
 - szyfrowanie, 335, 393
 - dane, 401
 - dane zapisywanych na serwerze WWW, 399
 - hasła, 378
 - z kluczem prywatnym, 336
 - z kluczem publicznym, 337
 - z kluczem tajnym, 336
- ## Ś
- ścieżka
 - bezwzględna, 80
 - względna, 80
 - śledzenie użytkownika, 479
 - śledzenie zakupów użytkownika
 - podczas przeglądania, 564
 - środowisko, 427
 - środowisko PHP, 496
 - środowisko programistyczne, 511
 - KPHPDevelop, 511
 - Zend Studio, 511
 - Świeża poczta, 599, 602
 - architektura skryptu, 604
 - baza danych, 603
 - dostosowywanie interfejsu użytkownika, 609
 - ekran logowania, 608
 - ilosc_kont(), 615
 - IMAP, 600
 - indeks.php, 604

konfiguracja bazy danych, 603
 konfiguracja kont, 611
 konta, 611
 logowanie, 608
 modyfikacja konta, 614
 Nowa wiadomość, 624
 odczytaj_wiadomosc(), 621
 odczytywanie wiadomości pocztowych, 615, 619
 odpowiadanie na wiadomość, 625
 otworz_skrzynke(), 618
 pliki aplikacji, 602
 pobierz_konta(), 612
 pobierz_liste_kont(), 617
 POP3, 600
 populacja.sql, 604
 potok IMAP, 619
 protokoły poczty, 599
 przechowaj_ustawienia_konta(), 613
 przeglądanie nagłówek wiadomości, 622
 przeglądanie zawartości skrzynki, 617
 przekazywanie poczty, 625
 sprawdz_uwierz_uzyt(), 610
 struktura systemu, 601
 stworz_baze.sql, 603
 szczegóły kont, 612
 tworzenie konta, 613
 usun_konto(), 614
 usun_wiadomosc(), 623
 usuwanie konta, 614
 usuwanie wiadomości, 623
 wybór konta, 615
 wylogowanie, 608
 wyslij_wiadomosc(), 624
 wyswietl_konf_konta(), 612
 wyswietl_liste(), 617
 wysyłanie wiadomości, 623

T

t1lib, 458
 tabele, 154, 208, 215
 aliasy, 248
 ARCHIVE, 298
 atrybuty, 208
 atrybuty kolumn, 228
 CSV, 298
 InnoDB, 295, 298
 klucze, 209, 214
 kolumny, 208
 krotki, 208
 ładowanie danych z pliku tekstowego, 297
 łączące tabele, 215
 maszyny zapisu, 298
 MEMORY, 298
 MERGE, 298
 MyISAM, 298, 301
 optymalizacja, 291

pola, 208
 puste pola, 214
 rekordy, 208
 relacje, 210
 tworzenie, 227
 typy kolumn, 229
 usuwanie, 257
 wartości, 208
 wiersze, 208
 złączenia, 245
 zmiana struktury, 255
 zmiana typu, 299
 tables_priv, 276, 278, 280, 281
 tablice, 97
 \$_COOKIE, 480
 \$_FILES, 414
 \$_GET, 48
 \$_POST, 48
 \$_REQUEST, 48
 \$_SESSION, 483
 dołączanie funkcji do każdego elementu, 115
 dostęp do elementów, 61, 99, 101
 elementy, 98
 foreach, 101
 indeks, 98, 99
 indeksy nienumeryczne, 100
 indeksy numeryczne, 98
 inicjowanie, 100
 kardynalność, 117
 klucz, 98, 99
 konwersja na zmienne skalarne, 117
 liczba elementów, 112
 ładowanie tablic z plików, 112
 modyfikacja zawartości elementów, 99
 niepowtarzalne wartości, 116
 odwracanie kolejności, 111
 operatory, 61, 103
 pętle, 100, 101
 poruszanie się wewnątrz tablicy, 114
 porządkowanie, 107
 przetwarzanie, 100
 sortowanie, 106
 sortowanie odwrotne, 107
 superglobalne, 48
 tworzenie, 98
 wysświetlanie zawartości, 99
 zliczanie elementów, 116
 zmiana kolejności elementów, 110
 tablice dyskusyjne, 671
 tablice tablic, 105
 tablice wielowymiarowe, 97, 103
 dostęp do danych, 104
 tablice, 108
 tworzenie, 103
 Tahuayo, 732, 734
 akcje, 741
 BUFOR, 739

- Tahuayo
 - buforowanie danych pochodzących z ządania, 756
 - do_bufora(), 757
 - funkcje_bufora.php, 757
 - funkcje_kategorii.php, 742
 - funkcje_koszyka.php, 759
 - funkcje_wysw_ksiazek.php, 743
 - IDENTPROG, 739
 - IDENTWSPOL, 739
 - indeks.php, 734
 - informacje na temat produktu, 751
 - instalacja kodu źródłowego, 762
 - kasa, 761
 - koszyk na zakupy, 741, 758
 - METODA, 739
 - odczytywanie wyników, 750
 - pliki aplikacji, 737
 - pobierz_wv(), 743, 756
 - połączenie z serwerem Amazon.com, 744
 - prezentacja listy kategorii, 742, 743
 - Produkt, 751
 - przejsie do kasy na witrynie Amazon.com, 761
 - przetwarzanie wyniku zapytania, 751
 - REST, 750
 - SOAP, 755
 - stale.php, 739
 - wczytaj_dokument_xml(), 751
 - węzeł przegladania, 740
 - wykonywanie ządania, 750
 - wyniki zapytania, 743
 - wyniki_wyszukiwania.php, 744
 - WynikiWyszukiwania, 743, 757
 - wyswietl_kategorie(), 742
 - wyswietl_koszyk(), 761
 - wyswietl_wp(), 743
 - wyszukaj_wp(), 748
 - wyszukiwanie według kategorii, 748
 - wyświetlanie listy ksiązek z danej kategorii, 742
 - z_bufora(), 757
- tar, 401
- TCP, 396
- TCP/IP, 328, 368, 396, 444
- technologia Ajax, 764, 768
- tekst, 462, 465, 696
 - pozycjonowanie, 470
 - wpisywanie do przycisku, 471
- tekst zaszyfrowany, 335
- telnet, 396
- tempnam(), 406
- Tera Term SSH, 439
- testowanie, 332, 515
 - GPG, 404
 - obsługa PHP, 800
- TEXT, 236, 237, 298
- text/plain, 416
- Thawte, 333, 339
- THEN, 307
- ThreadsPerChild, 363
- throw, 195, 197
- TIME, 236
- time(), 448
- TIMESTAMP, 234, 236
- TINYBLOB, 237
- TINYINT, 234
- TINYINT UNSIGNED, 230
- TINYTEXT, 237
- TLD, 138
- TLS, 398
- tłumienie błędów, 61, 264
- Top Level Domains, 138
- touch(), 425
- traceroute, 328
- track_errors, 61, 528
- transakcje, 215, 298, 299
 - ACID, 299
 - atomowość, 299
 - InnoDB, 300
 - izolacja, 299
 - rozpoczynanie, 300
 - spójność, 299
 - trwałość, 299
 - tryb automatycznego zatwierdzania, 300
 - wycofane, 299
 - wycofanie, 300
 - zatwierdzanie, 300
 - zatwierdzone, 299
- transmisja danych w internecie, 328
- Transmission Control Protocol, 396
- Transport Layer Security, 398
- trigger_error(), 529
- trim(), 121, 122, 263
- Tripwire, 330
- true, 51, 58, 72
- True Type Font, 468
- TrueType, 458, 465, 468
- trwałość, 299
- try, 195
- tryb automatycznego zatwierdzania, 300
- tryb otwarcia pliku, 79, 81
- tworzenie
 - aplikacje WWW, 504
 - baza danych, 220
 - brozury internetowe, 312
 - cookie, 480
 - dokumenty PDF, 700, 701
 - dokumenty PDF z szablonu, 710
 - dokumenty RTF, 707
 - egzemplarz klasy, 169
 - forum WWW, 671
 - indeksy, 232
 - internetowa baza danych, 217
 - kadr obrazka, 460
 - katalogi, 421
 - klasy, 178
 - kod łatwy w utrzymaniu, 506
 - kopia bezpieczeństwa, 436

kopia zapasowa, 342
 kopia zapasowa bazy danych MySQL, 292
 obiekt XMLHttpRequest, 770
 obiekty, 169
 obrazy, 460
 pliki, 425
 pliki RTF, 700
 procedury składowane, 302
 tabele, 227, 298
 tablice, 98
 tablice wielowymiarowe, 103
 wiadomości w formacie MIME, 668
 typy danych, 51
 Array, 52
 Boolean, 51
 ciągi znaków, 50
 Double, 51
 Float, 51
 Integer, 51
 mixed, 65
 MySQL, 233
 NULL, 52
 Object, 52
 PHP, 51
 resource, 52
 rzutowanie, 52
 siła typu, 52
 String, 51
 typy MIME, 416, 463

U

uaktualnianie
 oprogramowanie, 361
 system operacyjny, 369
 uasort(), 109
 ucfirst(), 125
 uchwyt, 166
 ucieczka od znaków, 43, 125
 ucwords(), 125
 uid, 424
 ujawnienie informacji poufnych, 327
 uksort(), 109
 umask(), 422
 umieszczanie tekstu w obrazku, 461
 UNICODE, 237
 unieważnianie, 175
 unieważnianie danych wynikowych, 355
 UNIX Epoch, 447
 UNIX_TIMESTAMP(), 450
 unlink(), 92, 407, 425
 UNLOCK TABLES, 295
 unserialize(), 495, 496
 unset(), 66, 483, 485
 UNSIGNED, 229, 233
 UNTIL, 306
 UPDATE, 223, 226, 255

UPLOAD_ERR_CANT_WRITE, 416
 UPLOAD_ERR_FORM_SIZE, 416
 UPLOAD_ERR_INI_SIZE, 416
 UPLOAD_ERR_NO_FILE, 416
 UPLOAD_ERR_NO_TMP_DIR, 416
 UPLOAD_ERR_PARTIAL, 416
 UPLOAD_ERROR_OK, 416
 upload_max_filesize, 412
 upload_temp_dir, 412
 upload_tmp_dir, 413, 418
 uprawnienia, 329, 399
 pliki, 424
 UPS, 343
 URL, 183, 380, 430, 435, 480, 481, 764, 784
 urlencode(), 380, 432
 uruchamianie
 polecenia na serwerze WWW, 425
 skrypty PHP w pliku, 499
 USAGE, 225
 USE, 227
 user, 276
 User-agent, 765
 USING, 248
 usługi sieciowe, 431, 725, 729
 Amazon.com, 730
 AWS, 731
 AWSZone, 731
 buforowanie, 732
 instalacja kodu źródłowego, 762
 NuSOAP, 732
 odpowiedzi REST, 731
 REST, 726
 schemat XML, 728
 SOAP, 725, 726, 729, 732
 Tahuayo, 732, 734
 wczytywanie dokumentów XML, 731
 WSDL, 730
 żądania SOAP, 729
 usort(), 108, 109, 110
 ustawianie
 cookies, 480
 typ zmiennej, 65
 zapomniane hasła, 551
 zmienne środowiskowe, 427
 usuwanie
 baza danych, 258
 funkcje składowane, 304
 identyfikator sesji, 483
 katalogi, 421
 pliki, 92, 425
 rekordy z bazy danych, 257
 tabele, 257
 zmienne sesji, 483
 usuwanie błędów, 517
 błędy logiczne, 523
 błędy programistyczne, 517
 błędy składni, 517

usuwanie błędów
 błędy wykonania, 519
 identyfikacja powodów błędów, 525
 interakcja z bazą danych, 521
 komunikaty o błędach, 518
 obsługa błędów, 529
 odczyt plików, 521
 połączenie z serwisami sieciowymi, 522
 poziomy zgłaszania błędów, 527
 sprawdzanie danych wprowadzonych przez użytkownika, 523
 wywołanie nieistniejących funkcji, 520
 wyzwalamie błędów, 529
 zapis plików, 521
 zmiana ustawień zgłaszania błędów, 528
 zmienne, 525

UTF-8, 355, 727

utrata danych, 328, 348

uwierzytelnianie, 327, 332, 334, 373, 374, 533
 identyfikacja użytkowników, 373
 implementacja, 534, 540
 implementacja własnej metody, 388
 kontrola dostępu, 374
 kontrola sesji, 485
 mod_auth, 383
 mod_auth_mysql, 386
 podstawowe uwierzytelnianie, 380
 protokół HTTP, 380
 przechowywanie haseł dostępu, 376
 szyfrowanie haseł, 378
 zakodowane uwierzytelnianie, 381
 zastrzeżenie wielu stron, 379

użyteczność, 333, 346

użytkownicy, 349

V

VALUES, 240
 var_export(), 192
 VARCHAR, 234, 237
 VBScript, 44
 VeriSign, 333, 339
 VirtualPC, 361
 VMWare, 361
 vprintf(), 124
 vsprintf(), 124

W

wadliwa dokumentacja, 331
 walidacja danych, 398
 warstwa SSL, 396
 wartości logiczne, 58
 wartości NULL, 230
 wartości zwracane podczas przypisania, 56
 warunek połączenia, 246
 warunki logiczne, 59

wątki, 671
 WBMP, 459
 wcinięcie kodu, 508
 wczytywanie dokumentów XML, 731, 751
 wdrażanie nowych wersji, 361
 Web 2.0, 763
 Web Services Description Language, 730
 Webalizer, 314
 WebTrends Analytics, 314
 weryfikacja danych wejściowych, 263
 weryfikacja tożsamości użytkownika, 374
 węzeł, 672, 673
 główny, 673
 potomny, 673
 przodek, 673
 węzły przeglądania, 732
 WHERE, 243
 operatory porównania, 244
 while, 72, 101
 WHILE, 306
 wiadomości e-mail, 121, 430
 format MIME, 668
 wiadomości o błędach, 151
 wiele do wielu, 210
 wielkość pliku, 92
 wielodziedziczenie, 177
 interfejsy, 177
 wielowątkowe grupy dyskusyjne, 671
 wiersz poleceń, 499
 wiersze, 208
 window.location, 616
 Windows, 368
 Windows NT, 327
 Wireless Bitmap, 459
 WITH GRANT OPTION, 222, 224, 225
 witryny WWW, 28
 właściciel skryptu, 497
 właściwości, 166
 włączanie plików, 142, 356
 Word, 696, 697
 Word 2003, 697
 wózek na zakupy, *Patrz* koszyk na zakupy
 wpisywanie tekstu do przycisku, 471
 wprowadzanie, 510
 WSDL, 729, 730, 756
 wskazywanie typu klasy, 186
 wstawianie danych do bazy, 267
 wstrzykiwanie kodu, 349
 WWW, 215
 wybór bazy danych, 226, 265
 wydajność, 333
 MySQL, 34
 PHP, 30
 Wydawca RFC, 429
 wyjątki, 195, 200
 blok catch, 195
 blok try, 195
 definiowanie przez użytkownika, 197

- Exception, 195, 196
 - obsługa, 195
 - stos wywołań, 197
 - throw, 195
 - zgłaszanie, 195
- wykonywanie
 - ciągi, 494
 - skrypty PHP w pliku, 499
 - zapytania do bazy danych z poziomu strony WWW, 262
- wykresy, 471
- wylogowanie, 548, 608
- wyprowadzanie, 510
- wrażenia regularne, 119, 127, 130, 133, 137
 - adres pocztowy, 138
 - domeny najwyższego poziomu, 138
 - dopasowywanie specjalnych znaków literowych, 136
 - dostosowywanie, 133
 - klasy znaków, 134
 - kotwiczenie, 136
 - MySQL, 244
 - odnajdywanie podciągów, 139
 - Perl, 133
 - podwrażenia, 135
 - podwrażenia policzalne, 135
 - POSIX, 133
 - powtarzalność, 135
 - rozdzielanie ciągów, 139
 - rozgałęzianie, 136
 - zbiory znaków, 134
 - znaki specjalne, 137
- wysyłanie plików, 411, 661
 - \$_FILES, 414
 - bezpieczeństwo, 413
 - file_uploads, 412
 - kod błędu, 416
 - kod HTML, 412
 - post_max_size, 412
 - problemy, 417
 - skrypt PHP, 413
 - typy MIME, 416
 - upload_max_filesize, 412
 - upload_temp_dir, 412
 - upload_tmp_dir, 413
 - ustawienia konfiguracyjne, 412
- wysyłanie
 - poczta elektroniczna, 121, 430, 623
 - zapytania do bazy danych, 265
- wyszukiwanie, 78
 - dane, 77
 - dane spełniające określone kryteria, 243
 - dane w bazie, 242
 - dane w wielu tabelach, 245
- wyświetlanie
 - grafika, 463
 - wartość atrybutów klasy, 192
 - zawartość katalogu, 418

- wywołanie
 - funkcje, 46, 149
 - funkcje składowane, 304
 - metody, 173
 - nieistniejące funkcje, 151, 520
 - operacje klas, 172
 - procedury składowane, 303
- wyzwalanie błędów, 529

X

- XHTML, 764, 766
- XML, 43, 701, 726, 765, 768
 - przeźrenie nazw, 728
- XML Namespaces, 728
- XML Schema, 728
- XML Style Sweet Transformations, 701
- XMLHttpRequest, 768, 770
 - abort(), 769
 - getAllResponseHeaders(), 769
 - getResponseHeader(), 769
 - onreadystatechange, 769
 - open(), 769
 - readyState, 769
 - responseText, 769
 - responseXML, 769
 - send(), 769
 - setRequestHeader(), 769
 - status, 769
 - statusText, 769
 - tworzenie obiektu, 770
- xor, 60
- XSLT, 701, 768
- XSS, 349

Y

- Yahoo! Mail, 599
- YEAR, 236

Z, Ż

- zabezpieczenia
 - kod źródłowy, 351
 - sieć, 367
 - transakcje, 391
 - witryny, 321
- zachęcanie gości do wyrażania opinii, 314
- zagrożenia, 347
- zainteresowanie efektami działania witryny WWW, 314
- ZakładkaPHP, 533
 - Ajax, 777
 - baza danych, 537
 - członek.php, 545
 - diagram systemu, 535
 - dodaj_zak(), 556, 785

ZakładkaPHP

- dodaj_zak.php, 555, 780, 781, 784
- dodajNowaZakladke(), 779
- dodajOdpowiedzZakladki(), 780
- dodawanie zakładek, 555, 778
- formularz_rejestracji.php, 540
- funkcje_prawid_dane.php, 543
- funkcje_url.php, 556
- funkcje_uwierz.php, 547
- funkcje_wyswietl.php, 539
- funkcje_zakladki.php, 538
- identyfikacja użytkowników, 534
- implementacja bazy danych, 537
- implementacja rekomendacji, 559
- implementacja uwierzytelniania, 540
- implementacja witryny, 538
- lacz_bd(), 545
- logowanie, 545
- logowanie.php, 538
- loguj(), 547
- nowa_rejestracja.php, 541
- nowe_ajax.js, 777
- nowy_as.css, 777
- odczytywanie zakładek, 555
- personalizacja, 534
- pliki aplikacji, 536
- pobierz_losowe_slovo(), 553
- pobierz_urle_uzyt(), 557
- pobierzZadanieXMLHTTP(), 778
- powiadom_haslo(), 554
- prawidlowy_email(), 543
- przechowywanie zakładek, 535
- rejestracja użytkowników, 540
- rejestruj(), 544
- rekomendacja.php, 559, 561
- rekomendacje, 559
- rekomendowanie zakładek, 535
- rekomenduj_urle(), 561
- schemat bazy danych, 537
- sesje, 548
- sprawdz_prawid_uzyt(), 548
- sprawdzanie danych użytkownika
 - w bazie danych, 547
- strona początkowa, 538
- tworz_naglowek_html(), 539, 778
- ustaw_haslo(), 552
- ustawianie zapomnianych haseł, 551
- usun_zak(), 559
- usun_zak.php, 558
- usuwanie zakładek, 557
- wylog.php, 548
- wylogowanie, 548
- wypelniony(), 543
- wyswietl_dodaj_zak_form(), 779
- wyświetlanie zakładek, 557
- zakładki, 778
- zapomnij_haslo.php, 552
- zmiana hasła, 549
- zmiana_hasla.php, 550
- zmien_haslo(), 551
- zakodowane uwierzytelnianie, 381
- zakończenie sesji, 485
- zakończenie wykonania skryptu, 495
- załadowane rozszerzenia, 496
- zamiana podciągów, 130, 132
- zamykanie
 - kursor, 307
 - plik, 86
 - połączenie FTP, 442
 - połączenie z bazą danych, 267
- zapewnianie bezpieczeństwa transakcji, 391
 - internet, 393
 - komputer użytkownika, 392
 - przebieg transakcji, 392
 - SSL, 395
 - system docelowy, 394
- zapisywanie
 - atomowe wartości kolumn, 213
 - dane do bazy, 240
 - dane w pliku, 77, 79, 84
 - zdarzenia, 340
- zapobieganie dziedziczeniu, 176
- zapomniane hasła, 551
- zaporą sieciową, 282, 341
- zaprzeczenie korzystania z usługi, 332
- zapytania, 265
- zarządzanie dużymi projektami, 503
- zarządzanie współbieżnymi zmianami, 510
- zarządzanie zmianami współbieżnymi, 510
- zarządzanie zmiennymi, 65
- zasada najmniejszego przywileju, 221
- zasięg, 156
 - funkcje, 156
 - globalny, 156, 157
 - zmiennie, 54
- zasoby, 52
- zaufanie, 317
- zawartość dynamiczna, 45
- Zend Accelerator, 515
- Zend Encoder, 515
- Zend Optimizer, 514
- Zend Studio, 511, 515
- Zend Technologies, 32
- ZEROFILL, 233
- zgłaszanie do katalogu stron, 433
- zgłaszanie wyjątku, 195
- zgłaszanie zmiennych sesji, 482
- zintegrowane środowisko programistyczne, 511
- zlib, 458
- zliczanie elementów tablicy, 116
- złamanie zabezpieczeń dostępu do serwera, 349
- złączenia, 245
- złodziejce sprzętu komputerowego, 350

zmiana

- grupa pliku, 424
 - hasło, 549
 - kolejność elementów tablicy, 110
 - przywileje, 281
 - rekordy w bazie danych, 255
 - struktura tabel, 255
 - środowisko wykonawcze, 497
 - uprawnienia pliku, 424
 - ustawienia zgłaszania błędów, 528
 - wielkość liter w ciągu, 125
 - właściciel pliku, 424
 - właściwości pliku, 424
- zmienne, 47, 50, 65
- globalne, 54, 156
 - lokalne, 156, 304
 - nazwy, 47, 51
 - przypisanie wartości, 51
 - reinterpretacja, 67
 - serializacja, 495
 - sesje, 480, 482, 564
 - skalarne, 97
 - sprawdzanie stanu, 66
 - sprawdzanie typów, 65
 - superglobalne, 54
 - ustawianie typów, 65
 - zasięg, 54, 156
- zmiennne formularza, 47
- zmiennne składowe, 166
- zmiennne środowiskowe, 427
- lista zmiennych, 428
 - ustawianie, 427
- zmiennne zmiennych, 53
- znacznik czasu Uniksa, 447, 448
- formatowanie, 450
- znaczniki PHP, 43
- styl ASP, 44
 - styl krótki, 43
 - styl SCRIPT, 44
 - styl XML, 43
- znak zakończenia wiersza, 303
- znaki tabulacji, 44
- zniszczenie danych, 328, 348
- zwiększanie wartości produktów i usług, 319
- zwracanie wartości z funkcji, 109, 160
- żądania
- HTTP, 764
 - SOAP, 729

PROGRAM PARTNERSKI

GRUPY WYDAWNICZEJ HELION



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

Zmień swoją stronę WWW
w działający bankomat!

Dowiedz się więcej i dołącz już dzisiaj!

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



PHP i MySQL

Tworzenie stron WWW

Vademecum profesjonalisty

Wydanie czwarte

Recepta na sukces w przypadku tworzenia profesjonalnych stron WWW jest niezwykle prosta: wystarczy ogromne możliwości PHP, niezrównana wydajność MySQL i wielka, skuteczna do pomocy społeczności skupiona wokół tego narzędzia. Wynik? Działająca realizacja zamówionych stron i aplikacji internetowych. Wszystko, czego Ci jeszcze trzeba, to fachowa wiedza pozwalająca wykorzystać ten potencjał!

PHP i MySQL to jedna z najlepszych, najpopularniejszych technologii do tworzenia rozwiązań internetowych, a książka, którą trzymasz w rękach, to czwarte wydanie bestsellerowego i kultowego już kompendium wiedzy na temat tych narzędzi. Dzięki niej zorientujesz się w każdym z aspektów wykorzystania PHP wraz z bazą danych MySQL. Poznasz język PHP, metody przechowywania i wyszukiwania danych oraz operacje na plikach. Dowiesz się, jak tworzyć uniwersalny kod i sprawnie pracować z bazą danych. Nauczysz się szybko lokalizować i rozwiązywać problemy oraz zapewnić bezpieczeństwo aplikacji, a także efektywnie ukoołować technologię AJAX.

Podrecznik zawiera także mnóstwo praktycznych przykładów, demonstrujących wykorzystanie PHP i MySQL do realizacji różnych zadań związanych z funkcjonowaniem dynamicznych witryn WWW.

Składowo 20 tytułów programów wydawniczych z grupy i więcej z 14 tytułami nowego wydawnictwa!

- Wybrane zalety PHP i MySQL
- Podstawy języka PHP
- Metody przechowywania i wyszukiwania danych
- Operacje na plikach
- Zastosowanie tablic
- Wykorzystanie wyrażen regularnych oraz operacje na ciągach znaków
- Tworzenie uniwersalnego kodu – pewne jego wykorzystanie
- Obiekty w PHP
- Obsługa wadłków i błędów
- Praca z bazą danych MySQL
- Nierównowagi połączenia z bazą i problemy PHP
- Administracja MySQL
- Zamknięta zgodzienia związane z bazą MySQL
- Zapewnienie bezpieczeństwa tworzącym rozwiązaniem
- Metody uwierzytelnienia przy użyciu PHP i MySQL
- Wykorzystanie protokołów i funkcji sieci
- Generowanie grafik
- Wykorzystanie sesji
- Obsługa poczty elektronicznej za pomocą PHP
- Użycie technologii AJAX



Hellion

Sprawdzić najnowszą promocję
www.hellion.pl/promocje
 Książki najchętniej czytane
www.hellion.pl/kategorie
 Zamów informacje o nowościach
www.hellion.pl/nowosci

Wydawnictwo Hellion
 ul. Koszowska 1c, 44-100 Gliwice
 tel.: 32 230 99 83
 e-mail: hellion@hellion.pl
<http://hellion.pl>

hellion.pl
 księgarnia
 internetowa

Nr katalogowy: **5881**



Księgarnia internetowa:
<http://hellion.pl>



Zamówienia telefoniczne:
0 801 339900



0 601 339900

Cena 109,00 zł

10091 978-83-246-3177-3



9 788324 631773

Informacyjna w największym wydawnictwie