

IDŹ DO

PRZYKŁADOWY ROZDZIAŁ



SPIS TREŚCI

KATALOG KSIĄŻEK

KATALOG ONLINE

ZAMÓW DRUKOWANY KATALOG

TWÓJ KOSZYK

DODAJ DO KOSZYKA

CENNIK I INFORMACJE

ZAMÓW INFORMACJE
O NOWOŚCIACH

ZAMÓW CENNIK

CZYTELNIA

FRAGMENTY KSIĄŻEK ONLINE

PHP i MySQL. Wprowadzenie

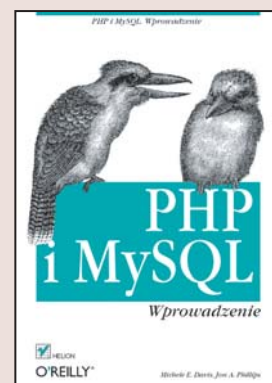
Autorzy: Michele Davis, Jon Phillips

Tłumaczenie: Radosław Meryk

ISBN: 83-246-0679-3

Tytuł oryginału: [Learning PHP and MySQL](#)

Format: B5, stron: 352



Twórz dynamiczne witryny WWW

- Poznaj język PHP
- Zaprojektuj w MySQL bazy i tabele
- Napisz komponenty witryn WWW
- Wprowadź mechanizmy zabezpieczeń

Styczne witryny WWW, które jeszcze całkiem niedawno królowały w internecie, dziś odchodzą do lamusa. Spośród milionów stron internetowych w pamięć internautów zapadają te, które „żyją”, są często aktualizowane, generowane dynamicznie i posiadają elementy interaktywne. Niestety, możliwości języka HTML nie pozwalają na tworzenie dynamicznych witryn WWW. W tym celu należy sięgnąć po inne technologie. Dziś najczęściej stosowanym rozwiązaniem jest połączenie strony internetowej z bazą danych oraz wykorzystywanie języka skryptowego, interpretowanego po stronie serwera, do pobierania treści z bazy i wyświetlania jej na stronie. Do takich zadań w większości witryn WWW używany jest język PHP i baza danych MySQL. Narzędzia te, poza ogromnymi możliwościami i niewielkimi wymaganiami sprzętowymi, mają ogromną zaletę – są dostępne nieodpłatnie.

Dzięki książce „PHP i MySQL. Wprowadzenie” zdobędziesz podstawową wiedzę na temat tworzenia dynamicznych witryn WWW za pomocą języka PHP i bazy danych MySQL. Dowiesz się, w jaki sposób zainstalować PHP i MySQL, jak przygotować pliki konfiguracyjne i dostosować działanie tych środowisk. Poznasz sposoby łączenia kodu PHP ze znacznikami HTML, elementy języka PHP i techniki programowania w nim. Nauczysz się projektować i tworzyć bazy danych oraz tabele, formułować zapytania w języku SQL i administrować serwerem baz danych. Połączysz skrypty PHP z bazą danych za pomocą biblioteki PEAR oraz napiszesz moduły obsługi formularzy, sesji i plików cookie. Przeczytasz także o zabezpieczeniu witryn WWW przed nieautoryzowanym dostępem.

- Zasady projektowania dynamicznych witryn WWW
- Instalacja PHP i MySQL
- Łączenie kodu PHP z HTML
- Struktury, obiekty i operatory języka PHP
- Projektowanie baz danych
- Podstawy tworzenia zapytań SQL
- Konfiguracja baz danych MySQL i zarządzanie tabelami
- Połączenie PHP i MySQL za pośrednictwem funkcji PHP i repozytorium PEAR
- Obsługa formularzy HTML w PHP
- Zarządzanie sesjami, plikami cookie
- Bezpieczeństwo PHP i MySQL

Poznaj technologię tworzenia „żywych” witryn WWW

Wydawnictwo Helion
ul. Kościuszki 1c
44-100 Gliwice
tel. 032 230 98 63
e-mail: helion@helion.pl



Spis treści

Przedmowa	7
1. Dynamiczna treść na stronach WWW	11
Miejsce technologii PHP i MySQL w procesie tworzenia aplikacji internetowych	11
Komponenty aplikacji PHP	13
Integracja wielu źródeł informacji	16
Żądanie danych ze strony WWW	20
Pytania do rozdziału 1.	22
2. Instalacja	23
Tworzenie aplikacji w trybie lokalnym	23
Praca zdalna	34
Pytania do rozdziału 2.	37
3. Poznajemy PHP	39
Obsługa tekstu w PHP	39
Cegielki kodowania	42
Pytania do rozdziału 3.	57
4. Podejmowanie decyzji w PHP	59
Wyrażenia	59
Operatory	61
Instrukcje warunkowe	67
Pętle	73
Pytania do rozdziału 4.	78
5. Funkcje	79
Wywoływanie funkcji	80
Definiowanie funkcji	82
Programowanie obiektowe	88
Pytania do rozdziału 5.	97

6. Tablice	99
Podstawowe informacje o tablicach	99
Pytania do rozdziału 6.	110
7. Podstawowe wiadomości o bazach danych	111
Projekt bazy danych	112
Strukturalny język zapytań	121
Pytania do rozdziału 7.	129
8. Korzystanie z MySQL	131
Baza danych MySQL	131
Zarządzanie bazą danych	134
Tworzenie kopii zapasowych danych i ich odtwarzanie	138
Zaawansowane instrukcje SQL	143
Pytania do rozdziału 8.	156
9. Komunikacja PHP z MySQL	159
Procedura	159
Wykonywanie zapytań do bazy danych za pomocą funkcji PHP	160
Korzystanie z PEAR	169
Pytania do rozdziału 9.	174
10. Obsługa formularzy	177
Tworzenie formularzy	177
Szablony	193
Pytania do rozdziału 10.	197
11. PHP w praktyce	199
Funkcje tekstowe	199
Funkcje przetwarzania dat i godzin	208
Operacje na plikach	212
Korzystanie z wywołań systemowych	221
Pytania do rozdziału 11.	222
12. Modyfikowanie obiektów MySQL i danych PHP	223
Modyfikowanie obiektów bazy danych z poziomu PHP	223
Wykonywanie operacji na danych tabeli	226
Wyświetlanie wyników z wykorzystaniem osadzonych łączy	226
Formularz do wprowadzania danych i jego obsługa w jednym pliku	229
Aktualizacja danych	234
Usuwanie danych	235
Wykonywanie podzapytań	239
Pytania do rozdziału 12.	240

13. Pliki cookie, sesje i kontrola dostępu	241
Pliki cookie	241
PHP i uwierzytnianie HTTP	244
Sesje	248
Wykorzystanie modułu Auth_HTTP do uwierzytniania	255
Pytania do rozdziału 13.	258
14. Bezpieczeństwo	259
Ograniczanie dostępu do stron administracyjnych	259
Bezpieczeństwo sesji	267
Pytania do rozdziału 14.	273
15. Sprawdzanie poprawności danych i obsługa błędów	275
Sprawdzanie poprawności danych wprowadzanych przez użytkowników za pomocą JavaScript	275
Dopasowywanie wzorców	279
Ponowne wyświetlenie formularza w przypadku niepomyślnego testu poprawności danych	282
Pytania do rozdziału 15.	285
16. Przykładowe aplikacje	287
Tworzenie blogu	287
Pytania do rozdziału 16.	308
17. Kończymy naszą podróż	311
Szukanie pomocy w internecie	311
Serwisy WWW	316
Zend	316
Lampshade	317
Pytania do rozdziału 17.	318
Dodatek Odpowiedzi na pytania do rozdziałów	319
Skorowidz	331

Dynamiczna treść na stronach WWW

Można wyróżnić dwa rodzaje stron WWW: statyczne i dynamiczne. W statycznych witrynach jest tekst zawierający hiperłącza, a czasami także ekran logowania, ale poza logowaniem użytkownik nie musi wykonywać żadnych dodatkowych działań. Przykładem statycznej witryny może być serwis <http://www.startribune.com>, w którym jedyną interakcją z użytkownikiem jest konieczność rejestracji w celu przeglądania artykułów. Z kolei witryna <http://www.amazon.com> jest przykładem dynamicznej witryny WWW. Dane o zamówieniach są rejestrowane, a kiedy użytkownik odwiedza stronę, witryna Amazon poleca określone pozycje na podstawie historii jego zakupów. Mówiąc inaczej, słowo „dynamiczne” oznacza, że komunikacja użytkownika z witryną nie ogranicza się do czytania stron — witryna zwraca informacje w odpowiedzi na działania użytkownika.

Jeszcze kilka lat temu tworzenie dynamicznych stron WWW wymagało pisania kodu o dużej objętości w językach C lub Perlu, a następnie wywoływania i wykonywania tych programów z poziomu procesu określanego jako *Common Gateway Interface* (CGI). Konieczność tworzenia plików wykonywalnych nie brzmi zachęcająco, podobnie jak nauka nowego, skomplikowanego języka. Na szczęście dzięki językowi PHP i bazie danych MySQL tworzenie dynamicznych witryn WWW jest dziś prostsze, łatwiejsze i szybsze.

Miejsce technologii PHP i MySQL w procesie tworzenia aplikacji internetowych

PHP jest językiem programowania przeznaczonym do interaktywnego generowania stron WWW na komputerze serwującym strony, zwanym *serwerem WWW*. W odróżnieniu od języka HTML, w którym przeglądarka WWW generuje strony na podstawie znaczników, kod PHP działa pomiędzy żądaniem strony a serwerem WWW, co pozwala na uzupełnianie i modyfikowanie kodu HTML. Na przykład kod PHP można wykorzystać do wyświetlania licznika odwiedzin na stronie.

Wystarczy mniej niż 20 wierszy kodu PHP, by zapisać adres IP, z którego pochodzi żądanie strony, w osobnym pliku, a następnie wyświetlić liczbę różnych adresów IP, z których żądano określonej witryny. Osoba żądająca strony WWW nie musi być świadoma, że stronę wygenerował interpreter PHP, ponieważ tekst licznika odwiedzin jest częścią standardowego kodu HTML wygenerowanego na podstawie kodu PHP.

Dzięki zastosowaniu języka PHP tworzenie stron WWW jest łatwe, ponieważ platforma PHP dostarcza całego potrzebnego kodu. Oznacza to, że za każdym razem, kiedy przystępujemy do

pisania programu w języku PHP, nie musimy na nowo „odkrywać koła”, tak jak w przypadku korzystania z języków kompilowanych, na przykład C.

Chociaż PHP doskonale nadaje się do generowania stron WWW, nie jest bazą danych. Programiści PHP zazwyczaj korzystają z bazy danych MySQL, której rolę można porównać do kancelisty rejestrującego informacje wprowadzane przez użytkowników. MySQL automatyzuje większość popularnych czynności związanych z zapamiętywaniem i odczytywaniem specyficznych informacji użytkowników na podstawie podanych kryteriów.



Na przykład w witrynie Amazon, w bazie danych MySQL mogą być zapisane rekomendacje serwisu wraz z informacjami o poprzednich zamówieniach.

Do bazy danych MySQL jest łatwy dostęp z poziomu kodu PHP. Technologie te często wykorzystywane są razem ze względu na ich bezproblemowe działanie. Dodatkową korzyścią jest fakt działania PHP i MySQL w komputerach różnych typów oraz pod kontrolą różnych systemów operacyjnych, między innymi Mac OS X, Windows i Linux.

Zalety wykorzystywania PHP z MySQL

Jest kilka powodów, dla których korzystanie z PHP wraz z MySQL jest naturalnym wyborem:

PHP i MySQL dobrze ze sobą współpracują

PHP i MySQL opracowano z zamiarem ich wspólnego wykorzystywania, dlatego używanie obu tych technologii razem nie sprawia kłopotów. Interfejsy programistyczne pomiędzy obydwojema produktami zostały logicznie dobrane. Wspólne wykorzystywanie obu technologii nie było pomysłem, który pojawił się już po utworzeniu interfejsów PHP i MySQL.

PHP i MySQL pozwalają na korzystanie z zalet oprogramowania typu open source

Ponieważ zarówno PHP, jak i MySQL są projektami typu *open source*, można z nich korzystać za darmo. Biblioteki klienckie obsługi MySQL nie są już dołączane do języka PHP. Zaawansowani użytkownicy mogą wprowadzać zmiany w kodzie źródłowym i w ten sposób modyfikować sposób działania języka i tworzonych za jego pomocą programów.

PHP i MySQL mają aktywne społeczności użytkowników

W internecie są aktywne społeczności użytkowników PHP i MySQL. Można stać się ich członkiem i uzyskać odpowiedzi na wiele pytań. Można również zakupić usługę profesjonalnej pomocy technicznej dla bazy danych MySQL, jeśli taka pomoc jest potrzebna.

PHP i MySQL pozwalają na tworzenie wydajnych aplikacji

Dzięki prostocie i dobremu projektowi aplikacje stworzone za pomocą PHP i MySQL działają szybko.

Użytkownicy PHP i MySQL nie muszą znać niepotrzebnych szczegółów

Dzięki standardowemu interfejsowi pozwalającemu na wywoływanie procedur obsługi bazy danych MySQL z poziomu języka PHP użytkownicy nie muszą znać niskopoziomowych szczegółów współpracy języka PHP z bazą danych MySQL. W witrynie <http://www.php.net> można znaleźć obszerny opis interfejsu API.

Zalety technologii open source

Jak wspominaliśmy wcześniej, zarówno PHP, jak i MySQL są projektami *open source*. Oznacza to, że aby z nich korzystać, nie trzeba kupować licencji dla każdego komputera w biurze czy też w domu. W projektach typu *open source* programiści mają dostęp do kodu źródłowego. Dzięki temu indywidualni programiści bądź ich grupy mogą analizować kod w celu wykrywania potencjalnych problemów, a także testować go, debugować oraz wprowadzać modyfikacje i ulepszenia. Na przykład system Unix — prekursor technologii *open source* — był udostępniany za darmo akademickiej społeczności specjalistów z dziedziny programowania. Linux — darmowa alternatywa systemu Unix — to bezpośredni efekt prac tych specjalistów i paradygmat licencji typu *open source*.

Jak powiedział Tim O'Reilly: „Początek licencji typu *open source* dały próby zachowania kultury udostępniania kodu, a dopiero później zdano sobie sprawę z wartości tego udostępniania”. Obecnie programiści systemów typu *open source* udostępniają sobie kod w internecie za pośrednictwem witryny *php.net*, serwerów list dyskusyjnych oraz innych witryn WWW. Jeśli kogoś dopadnie koszmar kodowania i nie może się z niego otrząsnąć, wymienione powyżej zasoby pomogą mu w przebudzeniu się i rozwiązaniu problemów.



W dalszej części książki wymienimy fora użytkowników technologii *open source* i zachęcimy Czytelników, by samodzielnie sprawdzili ich zalety. Wymienimy również serwery list dyskusyjnych i witryny WWW, z których warto skorzystać w przypadku problemów.

Komponenty aplikacji PHP

Aby można było tworzyć i rozwijać dynamiczne strony WWW, trzeba zapoznać się z kilkoma technologiami. Dynamiczne strony WWW wymagają zastosowania trzech głównych komponentów: serwera WWW, języka programowania po stronie serwera i bazy danych. Przed przystąpieniem do tworzenia aplikacji internetowych w PHP warto zapoznać się z tymi trzema podstawowymi komponentami projektowania aplikacji internetowych. Rozpoczniemy od podstawowych wiadomości na temat historii i przeznaczenia serwera Apache (serwera WWW), interpretera PHP (języka programowania po stronie serwera) i MySQL (bazy danych). Dzięki temu będziemy mogli łatwiej zrozumieć ich miejsce w procesie tworzenia aplikacji internetowych.

Należy pamiętać, że dynamiczne strony WWW pobierają informacje z kilku źródeł naraz, w tym z serwera Apache, interpretera PHP, bazy danych MySQL oraz kaskadowych arkuszy stylów (ang. *Cascading Style Sheets* — CSS). Wszystkie te technologie opiszemy w dalszej części książki.

Narodziny PHP

Inspiracją do powstania języka PHP była potrzeba tworzenia i utrzymywania witryn WWW zawierających dynamiczną treść i działających w architekturze klient-serwer. W 1994 roku Rasmus Lerdorf dla własnego użytku stworzył zbiór skryptów *open source* w Perlu. Ostatecznie skrypty te przepisano na język C, co dało początek językowi PHP. Do 1998 roku wydano PHP w wersji 3.x. System przekształcił się w narzędzie do tworzenia aplikacji internetowych, zdolne do konkurencji z takimi produktami, jak ASP (ang. *Active Server Pages*) firmy Microsoft czy też JSP (ang. *Java Server Pages*) firmy Sun.

Prawdziwa siła języka PHP polega na jego prostocie w połączeniu z dużymi możliwościami. Wielką zaletą jest również fakt, iż jest to język interpretowany, a nie kompilator.



Języki kompilowane tworzą binarne pliki wykonywalne (.exe), natomiast języki interpretowane bezpośrednio przetwarzają kod źródłowy i nie tworzą oddzielnych plików.

Język PHP jest dostępny dla wszystkich najpopularniejszych systemów operacyjnych. Jest łatwy do nauki, co czyni z niego idealne narzędzie dla osób stawiających pierwsze kroki w programowaniu aplikacji internetowych. Można również skorzystać z pracy innych programistów wchodzących w skład społeczności PHP, dzięki czemu tworzenie aplikacji internetowych staje się łatwiejsze dla każdego. Twórcy języka PHP opracowali infrastrukturę, która umożliwia doświadczonym programistom języka C rozszerzanie możliwości języka PHP. Dzięki temu język PHP integruje się z wieloma zaawansowanymi technologiami, takimi jak XML, XSL oraz COM firmy Microsoft. Do integracji tych technologii wykorzystuje się PHP w wersji 5.0.

Narodziny Apache

Apache jest serwerem WWW, który przekształca żądania serwera na wynikowe strony WWW oraz potrafi przetwarzać kod PHP. PHP jest jedynie językiem programowania, zatem bez działającego na zapleczu serwera WWW użytkownicy nie mieliby możliwości dotarcia do stron zawierających kod w języku PHP.

Apache nie jest jedynym dostępnym serwerem WWW. Innym popularnym serwerem WWW jest IIS (ang. *Internet Information Services*), dostarczany wraz z systemem Windows 2000 i wszystkimi nowszymi wersjami systemu Windows. Różnica pomiędzy systemem Apache a IIS w większości sprowadza się do osobistych preferencji, choć system Apache ma niewątpliwą przewagę pod tym względem, że jest darmowy, daje dostęp do pełnego kodu źródłowego i nie ma ograniczeń w licencji. Obecnie najnowsza wersja serwera Apache ma numer 2.0. Serwer IIS łatwiej integruje się z *Active Directory* — najnowszym systemem uwierzytelniania firmy Microsoft, ale w większości dotyczy to wewnętrznych witryn WWW działających w firmach.



Według ankiety w serwisie Netcraft serwer Apache od kwietnia 1996 roku jest najpopularniejszym serwerem WWW w internecie.

Ponieważ serwery WWW takie jak Apache i IIS zaprojektowano tak, by serwowały strony na podstawie plików HTML, muszą one zapewniać sposób przetwarzania kodu w języku PHP. W serwerze Apache wykorzystuje się mechanizm modułów, który umożliwia ładowanie rozszerzeń do funkcji podstawowych. W serwerze IIS wykorzystano podobny mechanizm określany jako ISAPI. Obie te technologie pozwalają na szybsze przetwarzanie kodu PHP w porównaniu ze starszym sposobem polegającym na wywoływaniu procesu PHP jako osobnego pliku wykonywalnego za każdym razem, kiedy serwer WWW zażąda strony zawierającej kod PHP. Sposób konfiguracji modułów Apache omówimy w rozdziale 2.

Wersje serwera Apache

Obecnie wykorzystywane są tylko dwie główne wersje serwera Apache. Są to wersje 1.3 i 2.0. Apache 2 to unowocześniona wersja, która obsługuje wielowątkowość. Wątki to mechanizm, który umożliwia pojedynczemu procesowi zarządzanie w określonym momencie więcej niż jedną operacją. Korzyść z zastosowania wątków polega na zwiększonej szybkości działania i mniejszej ilości potrzebnych zasobów. Niestety, do tej pory język PHP nie w pełni obsługuje wątki. Wersję Apache 2 opublikowano na tyle dawno, że można ją uznać za stabilną zarówno do celów projektowania, jak również do wykorzystania w rzeczywistych aplikacjach. W systemie Apache w wersji 2 poprawiono również obsługę modułów. Dodatkowe moduły można znaleźć pod adresem http://www.cri.ensmp.fr/~coelho/mod_macro/. Biblioteki DLL współdzielonych modułów, które nie są dostarczane z oficjalnymi plikami źródłowymi serwera Apache, takie jak *mod_php4*, *mod_ssl*, *mod_auth_mysql* oraz *mod_auth_ntsec*, również można znaleźć w internecie.

Narodziny MySQL

Bazę danych MySQL opracowano w latach pięćdziesiątych jako odpowiedź na rosnące potrzeby powstania narzędzia do inteligentnego zarządzania informacjami. Pierwsi twórcy bazy danych MySQL starali się spełnić swoje potrzeby w zakresie baz danych za pomocą niewielkiej i prostej bazy danych *mSQL*. Kiedy stało się jasne, że baza danych *mSQL* nie jest w stanie rozwiązać wszystkich problemów, które napotkali, opracowali bardziej rozbudowaną bazę danych, która przekształciła się w MySQL.

Baza danych MySQL obsługuje kilka różnych mechanizmów obsługi baz danych (ang. *database engine*). Mechanizm obsługi baz danych determinuje sposób, w jaki MySQL obsługuje zapisywanie danych oraz wykonywanie zapytań o dane. Z tego względu każdy mechanizm obsługi baz danych charakteryzuje się innym zestawem własności oraz silnych i słabych stron. Z czasem dostępne mechanizmy baz danych stają się coraz bardziej zaawansowane i działają szybciej. W tabeli 1.1 wyszczególniono numery wersji bazy danych MySQL wraz z ulepszeniami, które w nich wprowadzono.

Tabela 1.1. Główne wersje bazy danych MySQL

Wersja	Własności
3.23	Dodano mechanizm obsługi bazy danych MyISAM, który stał się mechanizmem domyślnym. Mechanizm ten pozwala na wydajną obsługę danych o dużej objętości. Wprowadzono mechanizm obsługi bazy danych InnoDB zapewniający bezpieczne przetwarzanie bazy danych oraz obsługę <i>obcych kluczy</i> . Obce klucze zapewniają jawne określenie związków pomiędzy tabelami na poziomie bazy danych.
4.0	Wprowadzono obsługę <i>unii</i> w zapytaniach. Unie umożliwiają scalanie wyników dwóch zapytań w jeden. Wprowadzono możliwość dokonywania zmian w konfiguracji bez konieczności ponownego uruchamiania bazy danych.
4.01	Dodano polecenie <code>help</code> do klienta pomocy. Wprowadzono obsługę widoków bez nazwy, określanych także jako <i>podzapytania</i> . Widoki bez nazwy pozwalają na traktowanie zapytania tak, jak osobnej tabeli.
5.0	Wprowadzono <i>wyzwalacze</i> , <i>procedury składowane</i> (ang. <i>stored procedures</i>) oraz <i>kursory</i> . Wyzwalacze umożliwiają wykonywanie kodu przez serwer bazy danych w momencie zajścia zdarzenia wyzwalającego, takiego jak wprowadzanie danych do tabeli. Procedury składowane umożliwiają definiowanie i wykonywanie programów w obrębie bazy danych. Kursory umożliwiają uruchamianie w bazie danych kodu dla każdego wiersza spełniającego warunki zapytania.
5.1	Wprowadzono obsługę <i>ograniczeń</i> i <i>partycjonowania</i> . Ograniczenia wykorzystuje się w celu zdefiniowania reguł, zgodnie z którymi można dodawać bądź modyfikować wiersze w bazie danych. Partycjonowanie wykorzystuje się w celu podzielenia fizycznej lokalizacji dużych tabel na podstawie określonej reguły. Partycjonowanie powszechnie stosuje się w celu zwiększenia wydajności dużych tabel, takich jak dane historyczne.

Bieżąca wersja „produkcyjna” bazy danych MySQL to najnowsze dostępne wydanie o numerze 5.0x. Wydajność serwera bazy danych MySQL 5.0 można porównać do znacznie droższych korporacyjnych systemów baz danych, takich jak Oracle, Informix, DB2 (IBM) czy też SQL Server (Microsoft). Twórcy bazy danych MySQL osiągnęli wydajność na tym poziomie dzięki wykorzystaniu talentów wielu programistów oprogramowania typu *open source*, a także testowaniu systemu przez społeczność użytkowników. W przypadku zadań obsługi baz danych ogólnego przeznaczenia domyślny mechanizm obsługi bazy danych — MyISAM — doskonale spełnia swoją rolę.



Najnowsze zaawansowane własności wprowadzone w MySQL 5.1 nie są równie stabilne, jak własności wprowadzone we wcześniejszych wersjach. Bieżącą stabilną wersją ogólnego przeznaczenia jest MySQL 5.0. Dla wybranej wersji głównej zawsze należy pobierać najnowsze wydanie (oznaczone najwyższym trzecim członem numeru wersji). Takie wydanie zawiera najwięcej poprawionych błędów określonej wersji.

Wielką zaletą serwera Apache jest jego zdolność do działania w systemach operacyjnych innych niż Windows. Działanie w różnych systemach operacyjnych wiąże się z problemem zgodności, który omówimy w kolejnym punkcie.

Zgodność

Przeglądarki WWW takie jak Firefox, Netscape i Internet Explorer przetwarzają kod HTML, zatem nie ma dla nich znaczenia, w jakim systemie operacyjnym pracuje serwer WWW. Systemy Apache, PHP i MySQL obsługują wiele systemów operacyjnych. Dzięki temu ich użytkownicy nie są skazani na korzystanie z określonego systemu operacyjnego ani po stronie serwera, ani po stronie klienta. Chociaż nie trzeba zbyt wiele się martwić problemami zgodności oprogramowania, należy się przyzwyczaić do występowania różnych formatów plików i wielu wersji językowych.

Integracja wielu źródeł informacji

W początkowych dniach internetu życie było proste. Były pliki zawierające kod HTML oraz pliki binarne, na przykład zawierające grafikę. Od tego czasu opracowano wiele technologii umożliwiających organizowanie wyglądu stron WWW. Na przykład *kaskadowe arkusze stylów* (ang. *Cascading Style Sheets* — CSS) pozwoliły na oddzielenie informacji opisujących prezentację od kodu HTML i umieszczenie ich w jednym miejscu. Dzięki temu raz wprowadzone zmiany obowiązują dla całego zbioru stron, co pozwala na uniknięcie konieczności ręcznej modyfikacji znaczników HTML w każdej ze stron z osobna.

W plikach HTML mogą jednocześnie występować odwołania do arkuszy CSS, szablonów PHP oraz bazy danych MySQL. Szablony PHP ułatwiają wprowadzanie zmian w kodzie HTML strony, jeśli zawierają pola wypełniane za pomocą zapytań do bazy danych. W kolejnych punktach omówimy każde z wymienionych wcześniej źródeł informacji.

Baza danych MySQL

MySQL jest systemem zarządzania relacyjnymi bazami danych. Dane nie są w nim zapisywane w jednej przestrzeni, ale w osobnych tabelach. Taka architektura zwiększa możliwości bazy danych oraz szybkość jej działania. Skrót SQL w nazwie MySQL pochodzi od *Structured Query*

Language (strukturalny język zapytań). Jest to najpopularniejszy wykorzystywany obecnie język obsługi baz danych stosowany w wielu systemach baz danych. W celu zaprezentowania postaci kodu obsługi bazy danych MySQL z poziomu języka PHP spójrzmy na listing 1.1, na którym z poziomu skryptu PHP wywołano kod usuwający dane użytkownika z bazy danych MySQL.

Listing 1.1. Funkcja PHP usuwająca dane użytkownika z tabeli bazy danych *user_name*

```
<?php

// Funkcja usuwająca użytkownika z tabeli site_user na podstawie
// parametru $user_name.
// Założono, że wcześniej otwarto połączenie z bazą danych.

function remove_user($user_name){
// Usunięcie użytkownika
// To jest instrukcja SQL
$sql_delete = "DELETE FROM `site_user` WHERE `User`='$user'";
$success = mysql_query($sql_delete) or die(mysql_error());

// wyświetlenie nagłówka strony
print('
<html>
  <head>
    <title>Usuwanie użytkownika</title>
    <link rel="stylesheet" type="text/css" href="user_admin.css" />
  </head>
  <body>
    <div class="user_admin">');
// Sprawdzenie, czy operacja usunięcia powiodła się
if ($success){
  // Poinformowanie użytkownika o sukcesie operacji
  print("Konto użytkownika $user_name pomyślnie usunięto.");
}
else {
  // Poinformowanie użytkownika o niepowodzeniu operacji
  print("Nie można usunąć konta użytkownika $user. Proszę spróbować później.");
}

  // Wyświetlenie stopki strony
  print('</div></body></html>');
}

?>
```

Nie przejmujmy się, jeśli niektóre fragmenty kodu na listingu 1.1 są niezrozumiałe. Ważne, jeśli zwrócimy uwagę, że w kodzie występują instrukcje PHP, kod dostępu do bazy danych oraz łącze do arkusza stylów.

Szablony PHP

W celu uproszczenia obsługi witryn składających się z wielu różnych stron o podobnym wyglądzie nagłówki i stopkę każdej ze stron można zdefiniować w osobnym pliku i włączyć do kodu każdej ze stron. Dzięki temu zmiany w nagłówku bądź stopce mogą być wykonywane w jednym miejscu i powodują automatyczną zmianę wyglądu stopki (nagłówka) na wszystkich stronach. W ten sposób programiści nie muszą modyfikować z osobna każdej ze stron witryny WWW.

Programiści PHP nauczyli się, że oddzielenie kodu PHP od kodu HTML ułatwia pracę zarówno im samym, jak i użytkownikom, którzy potrafią modyfikować kod HTML, ale niezbyt dobrze

rozumieją kod PHP. Dzięki utworzeniu osobnych plików szablonów zawierających zmienne-wypełniacze dla dynamicznych danych można oddzielić znaczniki HTML od kodu PHP.

Przykład pliku szablonu zamieszczono na listingu 1.2.

Listing 1.2. Szablon PHP

```
<html>
  <head>
    <title>Moje książki</title>
  </head>
  <body>
    <p>Ulubione książki:</p>
    <p>
      Tytuł: {$title}<br />
      Autor: {$author}
    </p>
  </body>
</html>
```

Kiedy interpreter PHP przetworzy stronę, zmienne-wypełniacze zostaną zastąpione ich wartościami, tak jak pokazano na listingu 1.3.

Listing 1.3. Wynikowy kod HTML po podstawieniu wartości za zmienne szablonu i przetworzeniu kodu przez interpreter PHP

```
<html>
  <head>
    <title>Moje książki</title>
  </head>
  <body>
    <p>Ulubione książki:</p>
    <p>
      Tytuł: Java. Almanach<br />
      Autor: Flanagan
    </p>
  </body>
</html>
```

Chociaż dodaliśmy kolejny plik, wynikowy kod HTML stał się czytelniejszy, a kod PHP mniej zaśmiecony nadmiarowym kodem HTML. Projektant witryny, który nie zna języka PHP, może zmodyfikować wygląd strony, nie obawiając się o to, że uszkodzi kod PHP.

Ostatni z rodzajów informacji, jakie omówimy, również powstał w efekcie dążenia do oddzielenia stylów prezentacji, takich jak kolory i odstępy, od zasadniczej treści.

Kaskadowe arkusze stylów

Kaskadowe arkusze stylów (ang. *Cascading Style Sheets* — CSS) wprowadzono do języka HTML po to, by projektanci aplikacji internetowych i ich użytkownicy mieli większą kontrolę nad sposobem wyświetlania stron WWW. Projektanci i użytkownicy mogą tworzyć arkusze stylów, które definiują sposób wyświetlania różnych elementów witryny, takich jak nagłówki i łącza. Pojęcie *kaskadowe* oznacza możliwość zastosowania wielu arkuszy stylów do tej samej strony WWW. W celu zastosowania kodu CSS w nagłówku pliku HTML należy wprowadzić kod o następującej postaci:

```
<html>
  <head>
    <title>Przykład zastosowania arkusza CSS</title>
    <style type="text/css">
```

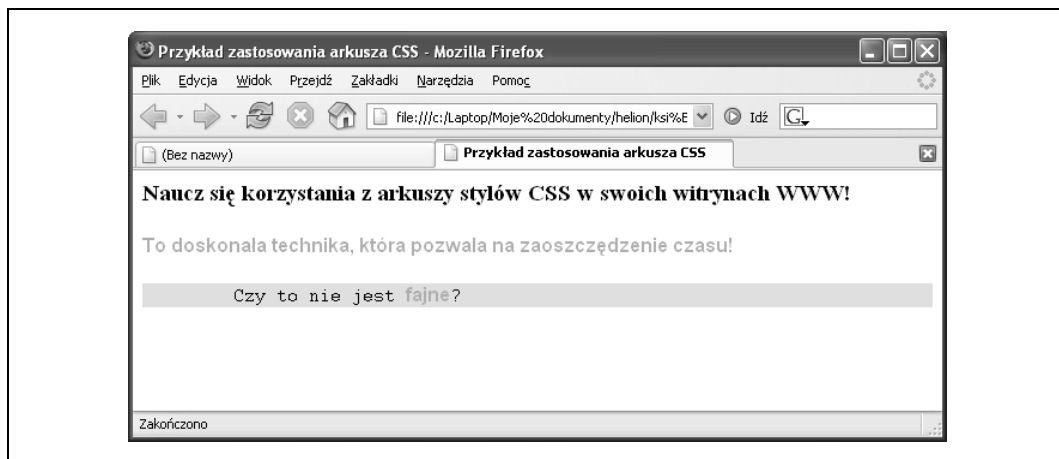
```

h4, b {color: #80D92F; font-family: arial; }
p { text-indent: 2cm; background: yellow; font-family: courier;}
</style>
</head>

<body>
  <h3>Naucz się korzystania z arkuszy stylów CSS w swoich witrynach WWW!</h3>
  <h4>To doskonała technika, która pozwala na zaoszczędzenie czasu!</h4>
  <p>Czy to nie jest <b>fajne</b>?</p>
</body>
</html>

```

Kod rozpoczynający się od znacznika `style` to kod CSS. Efekt wyświetlenia powyższego dokumentu w przeglądarce pokazano na rysunku 1.1.



Rysunek 1.1. Efekt wyświetlenia strony składającej się z kodu CSS i HTML w przeglądarce

Chociaż w pokazanym przykładzie kod CSS umieściliśmy wewnątrz pliku HTML, równie dobrze można by go było zapisać w osobnym pliku. Ten sposób wykorzystano na listingu 1.1, gdzie kod CSS pochodził z pliku `user_admin.css`.

Oczywiście do źródeł informacji wyświetlanych na stronach WWW trzeba zaliczyć także pliki zawierające standardowy kod HTML.



Więcej informacji na temat CSS można znaleźć w książce Erica Meyera *CSS. Kaskadowe arkusze stylów. Przewodnik encyklopedyczny* (Helion 2001).

Język znaczników HTML

Język znaczników HTML stosuje *znaczniki* do treści w celu zidentyfikowania informacji określonego typu bądź takich, które wymagają specjalnego formatowania. Znaczniki HTML zawsze są ujęte w nawiasy ostrokątne (`<>`), a wielkość liter nie ma w nich znaczenia, zatem nie jest ważne, czy znaczniki zapiszemy małymi, czy wielkimi literami (choć w specyfikacji języka XHTML zaleca się stosowanie małych liter). Znaczniki zazwyczaj składają się z par definiujących początek i koniec. Pary te mają następującą postać:

```
<znacznik>Czy to nie jest fajne?</znacznik>
```

Pierwszy <znacznik> oznacza początek pary, natomiast ostatni </znacznik> oznacza jej koniec. Parę znaczników określa się jako *element*. Dowolna treść w obrębie elementu podlega regułom obowiązującym dla tego elementu. W poprzednim przykładzie tekst *Naucz się korzystania z arkuszy stylów CSS w swoich witrynach WWW!* był zapisany wewnątrz elementu h3.

```
<h3>Naucz się korzystania z arkuszy stylów CSS w swoich witrynach WWW!</h3>
```

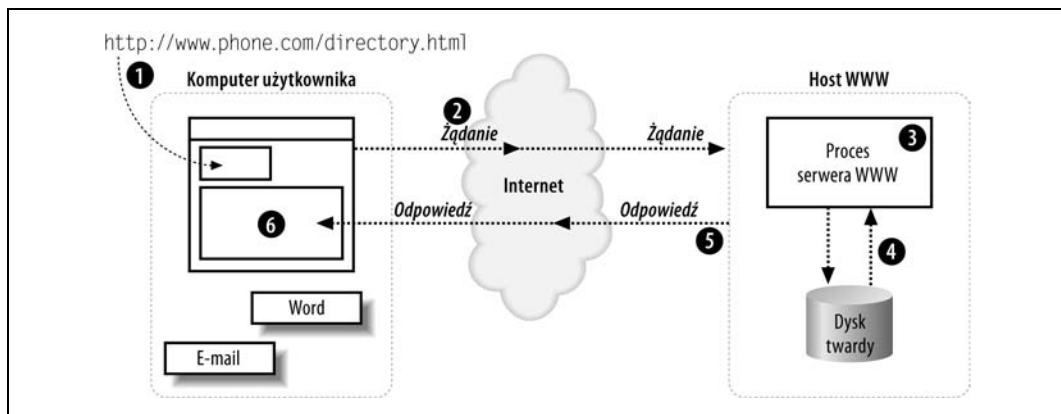
Dobrą praktyką (wymaganą w języku XHTML) jest dokładne zagnieżdżanie znaczników, dzięki czemu elementy mają dobrze zdefiniowane granice. Należy pamiętać o umieszczaniu znaczników końcowych elementów i unikać nakładania się znaczników na siebie. (Na przykład zamiast kodu w postaci pogrubienie <i>pogrubiona kursywa kursywa</i> należy stosować kod w postaci pogrubienie <i>pogrubiona kursywa</i> <i>kursywa</i>).

Żądanie danych ze strony WWW

Sposób integracji wymienionych powyżej części jest dość skomplikowany. Jeśli serwer WWW wykryje kod PHP, sprawdza, czy plik jest skryptem PHP, a jeśli tak, przekazuje żądanie przetwarzania strony do interpretera PHP. Od tego momentu w przetwarzaniu nie uczestniczy przeglądarka WWW. Jeśli jednak włączymy do pliku zewnętrzny plik CSS, przed wyświetleniem strony przeglądarka wysyła oddzielne żądanie dla tego pliku.

Interpretacja kodu PHP na serwerze

Interpretację kodu PHP na serwerze określa się terminem *przetwarzania po stronie serwera*. Żądanie strony WWW inicjuje cały szereg zdarzeń. Na rysunku 1.2 zilustrowano interakcję pomiędzy komputerem a serwerem WWW (hostem witryny WWW).



Rysunek 1.2. Kiedy użytkownik wpisze adres URL i wciśnie Enter, za kulisami wykonywanych jest szereg czynności w celu obsłużenia żądania

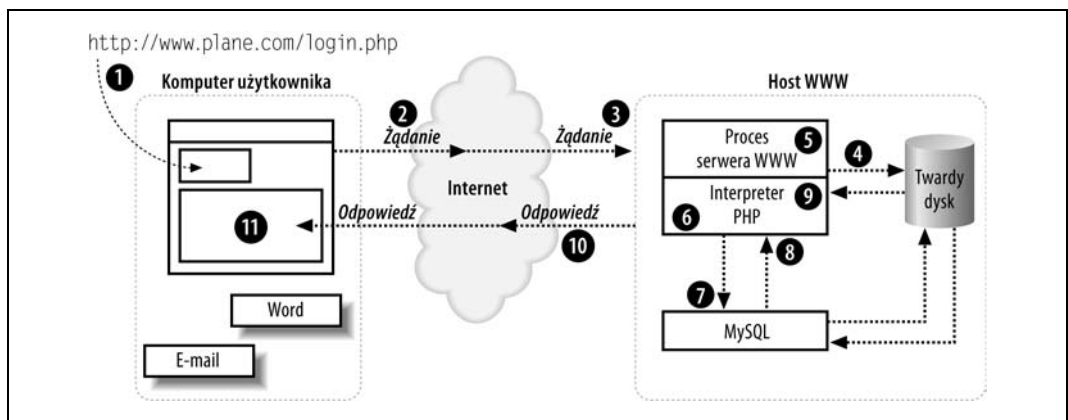
Oto analiza zdarzeń zilustrowanych na rysunku 1.2:

1. Użytkownik wprowadza adres strony w pasku adresu przeglądarki.
2. Przeglądarka analizuje adres i przesyła nazwę strony do serwera WWW. Na przykład wpisanie adresu URL `http://www.phone.com/directory.html` powoduje zażądanie strony `directory.html` z serwera `www.phone.com`.

3. Program na serwerze WWW określany jako proces serwera WWW pobiera żądanie strony *directory.html* i poszukuje tego pliku.
4. Serwer WWW odczytuje plik *directory.html* ze swojego twardego dysku.
5. Serwer WWW zwraca zawartość pliku *directory.html* do przeglądarki.
6. Przeglądarka wykorzystuje zbiór znaczników HTML odebrany z serwera WWW w celu wygenerowania strony WWW na ekranie komputera użytkownika.

Plik HTML o nazwie *directory.html* (którego żądanie zilustrowano na rysunku 1.2) określa się jako statyczną stronę WWW. Strona jest statyczna, ponieważ wszyscy użytkownicy, którzy zażąдают strony *directory.html*, otrzymują dokładnie taką samą stronę.

Dodanie do zestawu programów obsługujących żądania interpretera języka PHP i bazy danych MySQL umożliwia personalizację zwracanych stron. Na rysunku 1.3 pokazano dodatkowe operacje w ciągu zdarzeń zachodzących na serwerze WWW w przypadku zastosowania PHP i MySQL.



Rysunek 1.3. Współpraca interpretera PHP, bazy danych MySQL i serwera WWW podczas obsługi żądania strony

Oto operacje wchodzące w skład łańcucha zdarzeń:

1. Użytkownik wprowadza adres strony WWW w pasku adresu przeglądarki.
2. Przeglądarka analizuje adres i przesyła nazwę strony do hosta. Na przykład wpisanie adresu URL *http://www.phone.com/login.php* powoduje żądanie strony *login.php* z serwera *www.phone.com*.
3. Proces serwera WWW na hoście otrzymuje żądanie pliku *login.php*.
4. Serwer WWW odczytuje zawartość pliku *login.php* ze swojego dysku twardego.
5. Serwer WWW wykrywa, że plik PHP nie zawiera wyłącznie kodu HTML, zatem żąda przetworzenia pliku przez inny proces — interpreter PHP.
6. Interpreter PHP wykonuje kod PHP w kodzie otrzymanym od procesu serwera WWW. W kodzie PHP występują odwołania do bazy danych MySQL.
7. PHP żąda od procesu bazy danych MySQL przetworzenia odwołań do bazy danych.
8. Proces bazy danych MySQL zwraca wyniki zapytania do bazy danych.

9. Interpreter PHP uzupełnia wynik przetwarzania kodu PHP o dane z bazy danych i zwraca wyniki do procesu serwera WWW.
10. Serwer WWW zwraca wyniki do przeglądarki użytkownika w postaci kodu HTML.
11. Przeglądarka WWW użytkownika interpretuje odebrany kod HTML w celu wygenerowania strony na ekranie komputera użytkownika.

Może się wydawać, że wymienionych czynności jest dużo, wszystkie one są jednak wykonywane automatycznie za każdym razem, kiedy użytkownik zażąda strony WWW zawierającej kod PHP.

Podczas tworzenia dynamicznych stron WWW wykorzystuje się różnorodne zmienne i komponenty serwera. Wszystkie one mają istotne znaczenie dla uzyskania atrakcyjnej i łatwej w obsłudze witryny WWW. Nadszedł czas, aby zainstalować trzy główne części potrzebne do praktycznego tworzenia dynamicznych stron. Mowa o Apache, PHP i MySQL.

Pytania do rozdziału 1.

Pytanie 1.1. Jakie trzy komponenty są potrzebne do tworzenia dynamicznych stron WWW?

Pytanie 1.2. Jaki mechanizm wykorzystuje serwer Apache do ładowania rozszerzeń?

Pytanie 1.3. Jaka jest bieżąca stabilna wersja PHP?

Pytanie 1.4. Co oznacza ciąg SQL w nazwie MySQL?

Pytanie 1.5. Do czego służą ostrokątne nawiasy (< >)?

Pytanie 1.6. Jakie zadanie spełnia interpreter PHP?

Odpowiedzi na te pytania można znaleźć w Dodatku.