

## 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

# Adobe Creative Suite 3 Web Standard. Tworzenie dynamicznych stron WWW. Warsztaty



Autor: Paweł Zakrzewski  
ISBN: 978-83-246-1133-1  
Format: B5, stron: 392  
Zawiera DVD

### Zmień statyczne strony WWW w atrakcyjne i dynamiczne witryny!

- Zaprojektuj grafikę i strukturę witryny
- Napisz kod dynamicznych stron i skorzystaj z gotowych elementów
- Wzbogać witrynę elementami wykonanymi w technologii Flash i Ajax

Kto w dobie tętniących życiem, funkcjonalnych witryn chce dziś odwiedzać nudne, statyczne strony? Statyczne witryny WWW, nawet te najbardziej atrakcyjne pod względem treści i grafiki, mają także inną wielką wadę – ich aktualizacja zajmuje sporo czasu i wymaga „ręcznej” modyfikacji każdej ze stron. Wady tej pozbawione są witryny dynamiczne, w których treść przechowywana jest w bazie danych i pobierana przy każdym otwarciu przez odwiedzającego. Modyfikacja treści takiej witryny opiera się na wygodnym module administracyjnym, którego obsługa nie wymaga znajomości języka HTML.

Książka „Adobe Creative Suite 3 Web Standard. Tworzenie dynamicznych stron WWW. Warsztaty” to podręcznik dla tych twórców witryn WWW, którzy chcą przekształcić statyczne strony WWW w dynamiczne, korzystając z doskonałych narzędzi firmy Adobe, takich jak Dreamweaver CS3 i Fireworks CS3, oraz technologii PHP i MySQL. Czytając ją, poznasz zasady przygotowywania grafiki sieciowej, projektowania struktury dynamicznej witryny WWW i nawiązywania połączeń z bazą danych. Dowiesz się, w jaki sposób umieszczać na stronie elementy dynamiczne z biblioteki Dreamweavera, tworzyć strony z nowościami, moduły wyszukiwawcze i moduły zarządzania treścią. Znajdziesz tu także informacje o dodawaniu do strony elementów zrealizowanych w technologiach Flash i Ajax oraz o publikowaniu witryny na serwerze.

- Przygotowanie elementów graficznych w Fireworks CS3
- Projektowanie struktury witryny
- Instalacja środowiska PHP i bazy danych MySQL
- Tworzenie strony startowej
- Korzystanie ze skryptów znajdujących się w bibliotece Dreamweavera
- Implementacja mechanizmu wyszukiwania informacji w serwisie
- Moduł administracyjny
- Dodawanie elementów utworzonych za pomocą Flasha
- Przeniesienie witryny na serwer WWW

**Wykorzystaj możliwości Adobe Creative Studio 3  
i pracuj jak profesjonalista**

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



# Spis treści

<b>Informacje o autorze .....</b>	<b>5</b>
<b>Wstęp .....</b>	<b>7</b>
Informacje o autorze .....	7
Do kogo adresowana jest książka? .....	8
Co znajdziemy w tej książce? .....	8
Czego nie znajdziesz w tej książce? .....	11
<b>Rozdział 1. Grafika internetowa .....</b>	<b>13</b>
Planowanie .....	13
Rola grafiki na stronie WWW .....	13
Planowanie serwisu WWW .....	13
<b>Rozdział 2. Budowa podstawowej strony WWW .....</b>	<b>49</b>
Struktura serwisu WWW .....	49
Opracowanie struktury katalogów i definicja Site'a .....	49
Tworzymy statyczną stronę WWW .....	55
<b>Rozdział 3. Baza danych .....</b>	<b>89</b>
Środowisko pracy .....	89
Baza danych MySQL i PHP .....	89
<b>Rozdział 4. Dynamiczny serwis WWW .....</b>	<b>133</b>
Połączenie z bazą danych .....	133
Tworzenie dynamicznej wersji serwisu .....	133
Dynamiczna strona WWW .....	141
Edycja strony startowej .....	141
Przygotowanie źródła danych — Recordset .....	143
Dynamiczne menu na stronie .....	148
Dynamiczna zawartość strony startowej .....	158
Wprowadzenie elementów graficznych .....	162
Aktualności na stronie index.php .....	164
Formatowanie sekcji nowości .....	168
Wyświetlenie pełnej sekcji nowości .....	175
Sortowanie nowości .....	175
Warunkowe użycie elementów graficznych .....	176
Instrukcje warunkowe .....	177

<b>Rozdział 5. Dynamiczny serwis WWW .....</b>	<b>181</b>
Budowa dynamicznych stron WWW .....	181
Tworzenie strony Archiwum nowości .....	181
Budowa strony Informacje .....	191
<b>Rozdział 6. Wyszukiwanie danych .....</b>	<b>235</b>
Wyszukiwarki na stronie WWW .....	235
Tworzenie prostej wyszukiwarki .....	235
<b>Rozdział 7. Moduł zarządzania treścią .....</b>	<b>273</b>
Aktualizacja serwisu .....	273
Tworzenie systemu zarządzania treścią .....	273
Logowanie i zabezpieczanie .....	317
<b>Rozdział 8. Flash i Ajax na stronie WWW .....</b>	<b>325</b>
Animacje, efekty i funkcjonalność .....	325
Wykorzystanie technologii Flash .....	325
Budowa elementów animowanych SWF .....	326
Dynamiczny baner Flash .....	349
Wykorzystanie technologii Ajax .....	358
<b>Rozdział 9. Ostatni szlif, współpraca z serwerem WWW, dodatki.....</b>	<b>369</b>
Ostatnie poprawki.....	369
Testowanie .....	369
Eksport zawartości bazy danych .....	372
Co więcej?.....	379
<b>Skorowidz .....</b>	<b>383</b>

## Rozdział 4.

# Dynamiczny serwis WWW

## Połączenie z bazą danych

### Tworzenie dynamicznej wersji serwisu

W pierwszej części niniejszego opracowania zajęliśmy się planowaniem, a następnie przygotowaniem podstawowych elementów struktury naszej strony. W rezultacie przygotowane zostały konieczne do jej budowy elementy graficzne, katalog i definicja *site'a*, strona główna naszego serwisu — *index.html*, a także konieczne do jej atrakcyjnej prezentacji style kaskadowe. Nieco później przekształciliśmy główną stronę serwisu w szablon programu Adobe Dreamweaver CS3 i na tym zakończyliśmy dotychczasowe działania. Najwyższa pora, by przejść do najciekawszego etapu budowy dynamicznych serwisów — przekształcenia jego statycznej wersji w nową strukturę, opartą na bazie danych i kodzie PHP.

### Przygotowanie środowiska testowego w programie Adobe Dreamweaver CS3

Do budowy dynamicznej strony internetowej konieczne jest użycie testowego serwera WWW oraz odpowiednio skonfigurowanych baz danych i tabel MySQL. Aby uniknąć konieczności stałego użycia serwera WWW w Internecie, obciążenia łącza internetowego, a także dla szybkości działania i testów, najwygodniejszym rozwiązaniem jest wykorzystanie poznanego w poprzednim rozdziale serwera Krasnal (lub oczywiście dowolnego innego pakietu *serwer baza danych parser języka PHP*) umożliwiającego uruchamianie skryptów PHP lokalnie na dysku naszego komputera. Działania związane z instalacją serwera i konfiguracją baz danych za pomocą panelu *phpMyAdmin*, które podjęliśmy już w poprzednim rozdziale, pozwalają nam śmiało wkroczyć na dalszy etap pracy.

Uruchamiamy serwer testowy Krasnal.

## Edycja site'a

Do dalszego etapu budowy i testowania dynamicznej strony WWW nie wystarczy jedynie uruchomienie lokalnego serwera testowego. Konieczne jest także odpowiednie skonfigurowanie programu tak, by nie tylko pozwalał na pracę z plikami PHP, ale także ułatwiał ich przygotowanie oraz późniejszą edycję.

W tym celu musimy nieco zmodyfikować definicję całego *site'a* oraz zmienić lokalizację katalogu głównego strony. Działania podjęte w tej części rozdziału być może wywołają pewne zdziwienie u Czytelników i zrodzą kilka logicznych pytań.

W jakim celu budowaliśmy statyczną wersję strony, którą teraz będziemy musieli modyfikować? Czemu podczas definiowania struktury *site'a* w panelu *Site Definition* nie wskazaliśmy od razu właściwej lokalizacji?

Odpowiedzi jest wiele.

- ◆ Zwykle, gdy budujemy prototyp strony, wygodniej jest skorzystać z języka HTML i bez konieczności użycia serwera WWW testować stronę w różnych przeglądarkach za pomocą własnego komputera.
- ◆ Często trudno jest zacząć budowę bazy danych, gdy nie znamy jeszcze szczegółowych potrzeb klienta. By lepiej wykorzystać nasz czas, rozpoczynamy pracę od budowy statycznej strony, która ułatwia ocenę projektu graficznego i logiki nawigacji.
- ◆ Na etapie planowania i budowy statycznej wersji strony nie mieliśmy jeszcze dostępu do serwera testowego i nie było możliwości wskazania od razu odpowiednich danych. W tym momencie, dysponując już odpowiednim środowiskiem pracy, możemy dowolnie zmodyfikować ustawienia i dostosować je do naszych potrzeb.
- ◆ Być może podczas rozpoczynania budowy serwisu nie planowaliśmy jeszcze użycia bazy danych oraz skryptów języka PHP. Pomysł budowy dynamicznej strony zrodził się nieco później i wymaga teraz dostosowania naszej pracy do nowych założeń.
- ◆ Może już podczas tworzenia kolejnych stron statycznego serwisu doszliśmy do przekonania, że budowa następnych stron na bazie podobnego wzorca jest szalenie żmudna, monotonna i nudna. Mogłaby zostać w jakiś sposób zautomatyzowana i przyspieszona. Także i w takim przypadku odpowiedzią na nowe potrzeby jest właśnie przekształcenie bieżącego (istniejącego już) serwisu w dynamiczną postać przy użyciu bazy danych i skryptów PHP.
- ◆ Podczas (nawet zaawansowanej już) pracy nad serwisem pojawiła się konieczność dodania dodatkowej funkcjonalności niemożliwej do wykonania bez bazy danych i skryptów działających po stronie serwera. Dzięki reedycji ustawień struktury *site'a* możemy łatwo przejść do trybu budowy dynamicznej wersji strony, używając jako podstawy istniejącej już struktury.
- ◆ Być może nasz klient zamówił właśnie usługę przekształcenia bieżącej, statycznej wersji serwisu w nowoczesną dynamiczną stronę WWW, bez modyfikacji głównych elementów graficznych. W tej sytuacji umiejętność przekształcenia stron HTML w dynamiczne PHP okaże się niezwykle cenną umiejętnością.

W każdym wymienionym powyżej przypadku umiejętność przekształcenia bieżącej wersji strony w jej dynamiczną postać może okazać się doskonałym rozwiązaniem nie tylko dla przyspieszenia pracy nad stroną, ale także dla rozwinięcia struktury serwisu i dodania jego dodatkowej funkcjonalności, czy po prostu rozwiązaniem umożliwiającym w ogóle wykonanie zlecenia.

Takie właśnie rozumowanie stało się przyczyną zastosowania przedstawionej kolejności wykonywania działań przyjętej na łamach tej książki.

### Przygotowanie struktury katalogów

Aby w dalszej pracy skorzystać z możliwości uruchamiania skryptów PHP, wszystkie pliki naszego serwisu muszą znaleźć się na testowym serwerze WWW. W tym celu przenosimy (lub kopiujemy) cały katalog naszej witryny — *siteRecenzje* — w nową, docelową lokalizację — katalog o ścieżce: *C:\sr\Apache\httpd\html*.



Aby możliwe było przeniesienie katalogu (w celu archiwizacji lub bezpieczeństwa możemy przenieść w miejsce docelowe jedynie kopię folderu *siteRecenzje*), konieczne jest wyłączenie programu Adobe Dreamweaver CS3.

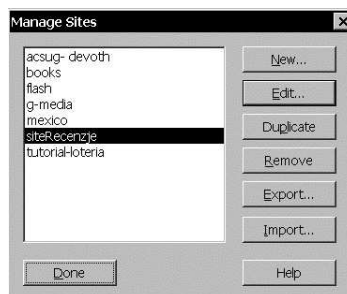
Jak pamiętamy, katalog o nazwie *html* jest głównym folderem naszego serwera testowego — *localhost*. Tu właśnie zlokalizowane powinny być wszystkie nasze dynamiczne aplikacje oraz elementy wykorzystywane do ich budowy.

Jeśli zdecydujemy się na przeniesienie (bez kopiowania) katalogu naszej strony, bezpośrednio po otwarciu programu Adobe Dreamweaver CS3 wyświetlony zostanie komunikat ostrzegawczy o braku folderu zawierającego strukturę naszej witryny. Aby niezwłocznie przejść do trybu edycji ustawień *site'a*, wybieramy opcję *Manage Sites* w wyświetlonym oknie dialogowym. Edycji ustawień *site'a* możemy dokonać także później, gdy zdecydujemy się na użycie opcji *Cancel*.

### Edycja właściwości site'a

Jeśli podczas otwierania Adobe Dreamweavera CS3 od razu zdecydujemy się na przejście w tryb edycji *site'a* za pomocą opcji *Manage Sites*, wyświetlone zostanie okno dialogowe z listą wszystkich serwisów budowanych na naszym komputerze. Aby przejść w tryb edycji naszej witryny, lokalizujemy i zaznaczamy jej nazwę w oknie po lewej stronie, a następnie za pomocą przycisku *Edit* wchodzimy tryb edycji *site'a* (rysunek 4.1).

**Rysunek 4.1.**  
*Okno Manage Sites przechowuje informacje o wszystkich serwisach, jakie budujemy za pomocą Dreamweavera. Aby łatwiej identyfikować każdą pracę, dobrze jest stosować unikalne i opisowe nazwy serwisów*

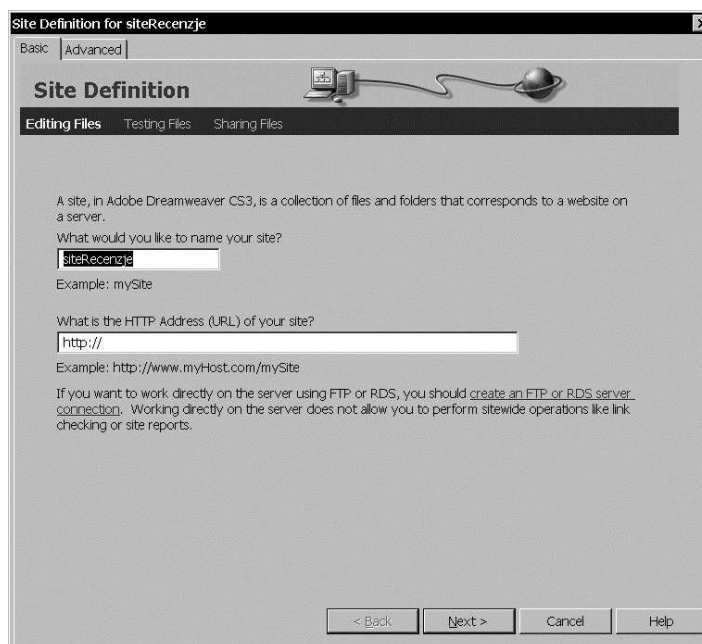


Jeśli zdecydowaliśmy się na utworzenie kopii serwisu w nowej lokalizacji lub też podczas otwierania programu wybraliśmy opcję *Cancel*, Adobe Dreamweaver CS3 otwiera się, nie wyświetlając żadnych komunikatów.

W celu przejścia do modyfikacji właściwości naszej strony wybieramy polecenie *Manage Sites* dostępne w menu górnym *Site*. W dostępnym oknie dialogowym wskazujemy nazwę naszego serwisu i za pomocą przycisku *Edit* wchodzimy w tryb edycji jego właściwości — *Site Definition* (rysunek 4.2).

#### Rysunek 4.2.

Za pomocą przycisku *Edit* w oknie *Manage Sites* przechodzimy w tryb edycji właściwości wybranego serwisu. W tym miejscu mamy możliwość modyfikacji wszystkich jego parametrów



Zakładka *Basic* widoczna w górnej części okna *Site Definition* pozwala na edycję ustawień za pomocą kreatora. Jest on niezwykle wygodnym narzędziem do zarządzania właściwościami witryny, szczególnie dla osób o mniejszym doświadczeniu.

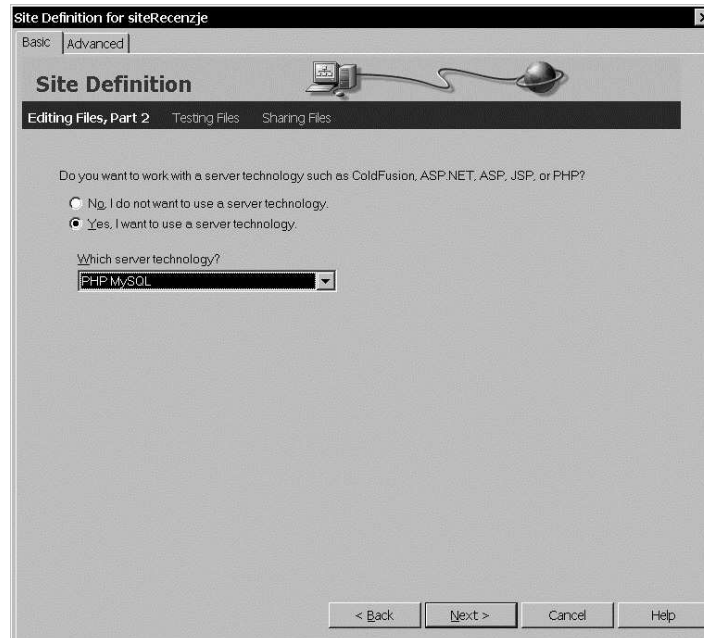
Na pierwszym ekranie kreatora *Site Definition* — *Editing Files* widzimy wprowadzoną wcześniej nazwę serwisu i w tym momencie nie musimy wprowadzać tu żadnych zmian. Za pomocą przycisku *Next* przechodzimy do ekranu *Editing Files, Part 2*

W tym miejscu musimy wprowadzić kilka istotnych zmian. Na pytanie *Do you want to work with a server technology such as ColdFusion, ASP.NET, ASP, JSP, or PHP?* wskazujemy odpowiedź *Yes, I want to use a server technology*, a następnie wybieramy *PHP/MySQL* z dostępnej poniżej listy — *Which server technology?* (rysunek 4.3).

Za pomocą przycisku *Next* przechodzimy do trzeciego okna, *Editing Files, Part 3*. Pytanie *How do you want to work with your files during development?* pozwala wybrać sposób, w jaki będziemy budować i edytować nasze strony w dalszej części pracy. Dzięki użyciu dostępnego na naszym dysku serwera testowego (Krasnal) mamy możliwość wyboru pierwszej opcji *Edit and test locally*

**Rysunek 4.3.**

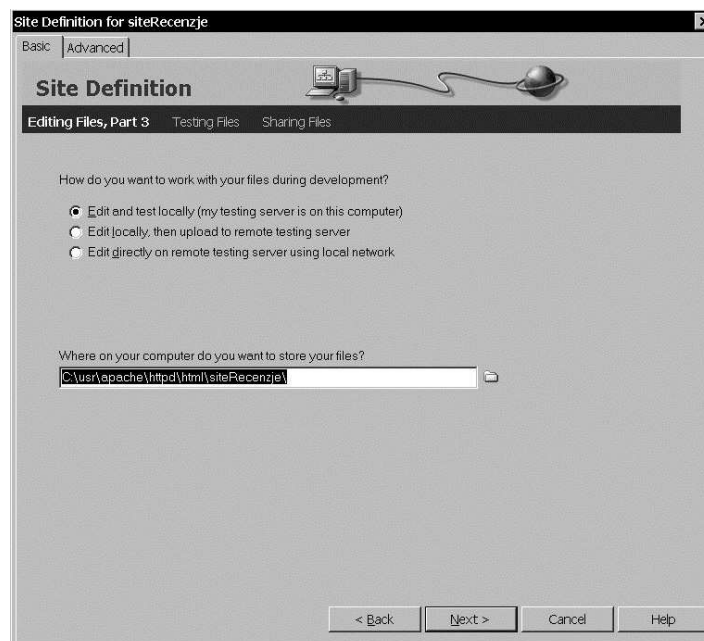
*Tym razem w oknie Editing File, Part 2 wskazujemy technologię, jaką wykorzystamy podczas budowy serwisu. Dreamweaver oferuje wsparcie i narzędzia do budowy dynamicznych aplikacji WWW dla wielu popularnych technologii — PHP+MySQL, ASP, JSP, Cold Fusion itp.*



*(my testing server is on this computer), która precyzyjnie określa, że do budowy i testowania dynamicznych stron użyjemy serwera testowego zainstalowanego na naszym komputerze. Zaznaczamy pierwszą opcję (rysunek 4.4).*

**Rysunek 4.4.**

*Opcja Edit and test locally pozwala nam na tworzenie serwisu lokalnie na dysku naszego komputera. Po zakończeniu pracy i testów, gotową i działającą stronę przesyłamy na serwer WWW*





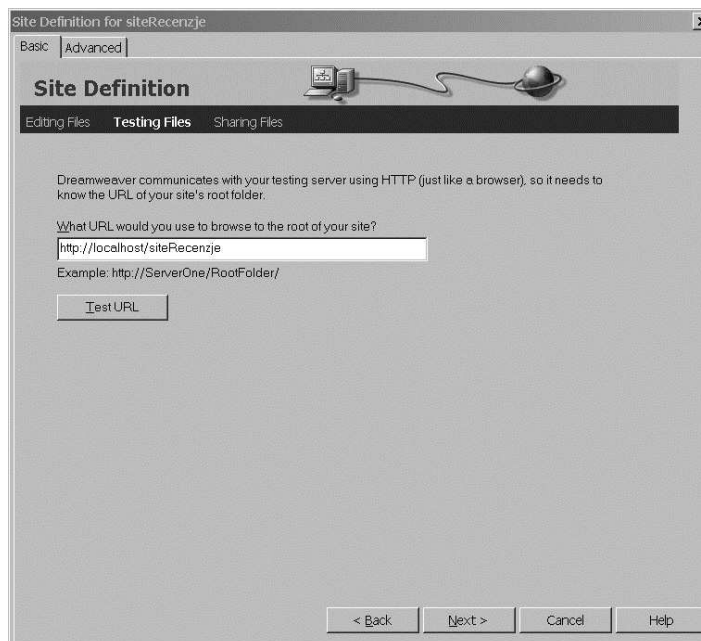
W polu poniżej, *Where on your computer do you want to store your files?*, należy wskazać bieżącą lokalizację głównego folderu naszej witryny — *C:\sr\Apache\httpd\html\siteRecenzje* (rysunek 4.5) i za pomocą przycisku *Next* przejść do sekcji *Testing Files*.

**Rysunek 4.5.**  
Widoczny w dolnej części okna zapis *Select:siteRecenzje* wskazuje docelową lokalizację testowej wersji naszej aplikacji na dysku lokalnego komputera



Na pytanie dotyczące adresu URL naszego serwisu — *What URL would you use to browse to the root of your site?*, w polu poniżej musimy na końcu widocznej ścieżki (*http://localhost*) wprowadzić nazwę — *siteRecenzje* (rysunek 4.6). To jest bowiem główny folder naszego *site'a* zapisany na lokalnym serwerze WWW. Jak pamiętamy, główny katalog serwera to inaczej *localhost*. W tym folderze znajduje się między innymi także nasz folder — *siteRecenzje*. W tej sytuacji ścieżka dostępu do witryny ma postać następującą: *http://localhost/siteRecenzje* i w tej formie wprowadzamy ją w polu *What URL would you use to browse to the root of your site?*

**Rysunek 4.6.**  
*localhost* to adres naszego serwera WWW Dynamiczny serwis z recenzjami książek zlokalizowany jest w katalogu *siteRecenzje*. Adres w postaci *localhost/siteRecenzje* wskazuje na stronę główną — *index.php* naszej aplikacji



Przycisk *Test URL* pozwala już na etapie definiowania ustawień *site'a* sprawdzić, czy katalog o wprowadzonej nazwie rzeczywiście istnieje i jest poprawnie widziany przez program (rysunek 4.7).

**Rysunek 4.7.**

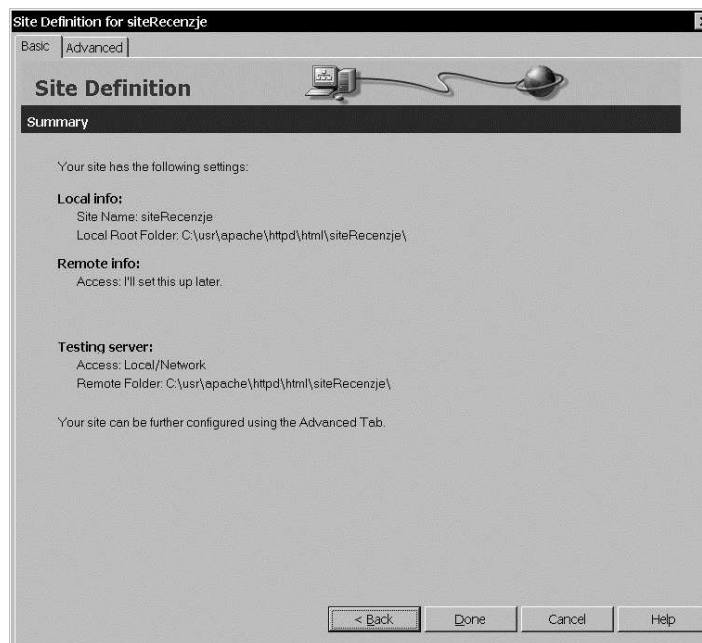
*Testowanie pozwala na sprawdzenie poprawności wprowadzonych adresów i lokalizowanie potencjalnych problemów*



Za pomocą przycisku *Next* przechodzimy do sekcji *Sharing Files*, gdzie nie wprowadzamy już żadnych modyfikacji i od razu przechodzimy dalej (*Next*). Widoczna sekcja *Summary* przedstawia podsumowanie wszystkich wprowadzonych i zmodyfikowanych ustawień naszego *site'a* i nie wymaga żadnych modyfikacji (rysunek 4.8). Za pomocą przycisku *Done* potwierdzamy zmiany i powracamy do okna edycji *Manage Sites*. Także w tym miejscu ponownie za pomocą przycisku *Done* zamykamy okno i jednocześnie potwierdzamy zmiany konfiguracyjne naszej strony.

**Rysunek 4.8.**

*W oknie podsumowania znajdziemy wszystkie podstawowe właściwości edytowanego serwisu*

**Zmiana typów plików**

Do budowy szablonu, a także pierwszej strony serwisu używaliśmy do tej pory klasycznych dokumentów HTML. Nadszedł czas, by przekształcić je w dynamiczne pliki PHP, które umożliwią interpretowanie kodu PHP bezpośrednio na serwerze WWW. Dzięki użyciu parsera (interpretacji) kodu PHP na serwerze otrzymamy w rezultacie prosty, statyczny dokument HTML w postaci czytelnej dla przeglądarki użytkownika.

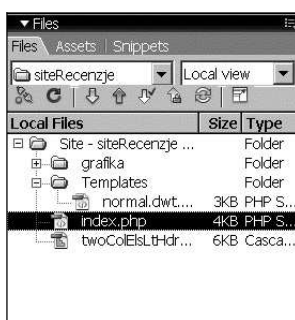
Zmiany rozpoczniemy od edycji szablonu. W tym celu otwieramy folder *Templates* i lokalizujemy dokument — *normal.dwt*. Do jego nazwy wprowadzamy dodatkowe rozszerzenie — *.php*. Nowa

nazwa szablonu — *normal.dwt.php* — wygląda nieco dziwnie, jednak gotowy dokument będzie działał poprawnie i jako wzorzec doskonale posłuży do budowy kolejnych stron PHP naszego serwisu.

W głównym katalogu *siteRecenzje* znajduje się ponadto pierwsza strona serwisu — *index.html*. Także ją powinniśmy przekształcić w dokument PHP. W tym celu w oknie *Files* wystarczy jedynie zmienić rozszerzenie pliku z *html* na *php*, nie wprowadzając już więcej dodatkowych zmian (rysunek 4.9).

#### Rysunek 4.9.

W celu przekształcenia statycznej wersji serwisu WWW w dynamiczną aplikację bazodanową konieczna jest zmiana rozszerzeń plików naszej strony. Mżemy wykonać to zarówno w oknie *Files* w Adobe Dreamweaverze, jak też systemowo w katalogu *siteRecenzje* na naszym dysku



Zmiany rozszerzeń plików szablonu — *normal.dwt* oraz strony głównej — *index.html* możemy dokonać w panelu *Files* programu Adobe Dreamweaver, ale także systemowo, w widoku podglądu zawartości folderu na dysku.

## Łączenie strony z bazą danych

Przygotowana wcześniej strona *index.php* jest podstawowym dokumentem naszej pracy, posłuży ona bowiem do budowy kolejnych stron serwisu. Zbudowana na podstawie szablonu — *normal.dwt.php* — zawiera zestaw stałych elementów (zablokowanych do edycji na stronach) oraz obszary edytowalne umożliwiające wprowadzanie odpowiednich treści na kolejnych stronach serwisu. Co ważne, edycja graficznej struktury szablonu — *normal.dwt.php* — pociągnie za sobą nie tylko zmianę startowej strony — *index.php*, ale także pozostałych podstron serwisu, podobnie jak *index.php* opartych na tym samym szablonie.

Analizując projekt strony *index.php* oraz zestaw przygotowanych wcześniej tabel MySQL, łatwo zauważyć, że boczne menu zawierające listę kategorii dostępnych książek powinno być generowane dynamicznie na podstawie danych z bazy. W ten sposób, dzięki użyciu dynamicznego menu na stronie szablonu, dodawanie kolejnych kategorii, ich edycja czy usunięcie będzie od razu widoczne na wszystkich stronach serwisu.



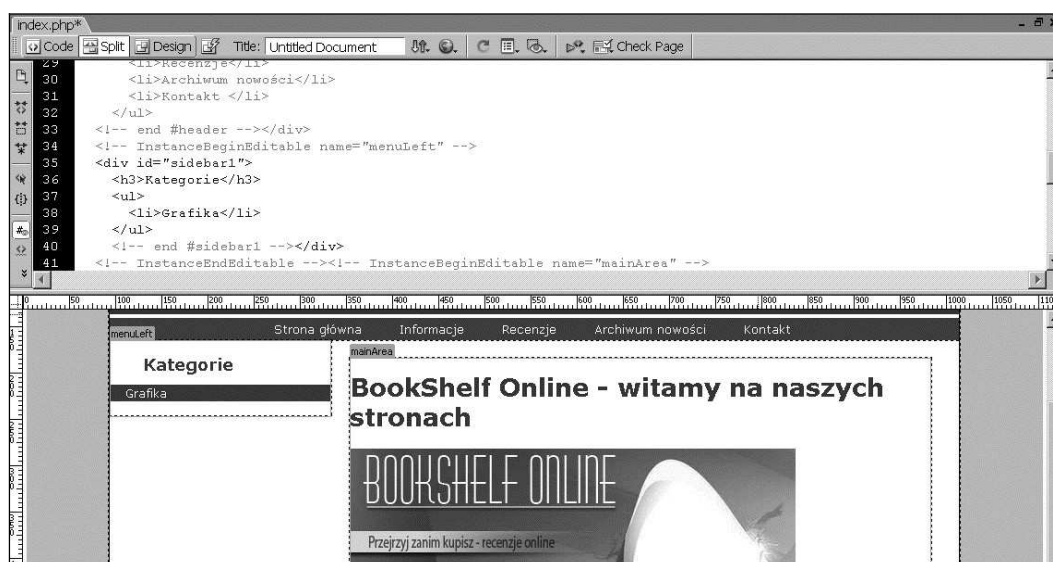
W przyszłości dzięki użyciu bazy danych i dynamicznej listy możliwe będzie także stosunkowo łatwe i szybkie przygotowanie dodatkowych wersji językowych czy innych modułów rozszerzających funkcjonalność całego serwisu.

## Dynamiczna strona WWW

### Edycja strony startowej

Jeśli decydujemy się na wykorzystanie listy kategorii książek na podstawie treści pobranych z bazy danych, statyczne, wprowadzone ręcznie nazwy kategorii widoczne w chwili obecnej nie są już potrzebne i w zasadzie mogą być usunięte. Jednak nie spieszymy się znowu, nie po to wprowadzaliśmy je wcześniej, by teraz w jednej chwili usunąć je bezpowrotnie. Aby jednak ułatwić sobie dalsze działania, wykorzystamy jedną, pierwszą pozycję listy do budowy nowej dynamicznej zawartości menu.

W tym celu uważnie, najlepiej w widoku kodu (widok *Source* lub *Split*), zaznaczamy kolejne nazwy kategorii, pozostawiając jedynie pierwszą pozycję, a następnie po prostu usuwamy zaznaczone elementy ze strony. W rezultacie pozostała nam jedynie nazwa jednej kategorii menu. Wykorzystamy ją jako wzorzec do dalszej pracy (rysunek 4.10).



**Rysunek 4.10.** W widoku kodu lokalizujemy i usuwamy nazwy kategorii. Do budowy dynamicznej wersji serwisu wystarczy nam jedna pozycja na liście

### Recordset — źródło dynamicznych danych

Do budowy dynamicznej strony WWW konieczne jest przygotowanie i dostarczenie jej odpowiednich danych. W programie Adobe Dreamweaver CS3 źródłem dynamicznych danych jest *Recordset*. Składa się on z kilku istotnych modułów przygotowanych za pomocą języka PHP i dołączanych do kodu HTML naszej strony. Każdy *Recordset* zawiera:

- ♦ Zestaw instrukcji umożliwiających nawiązanie komunikacji z bazą danych,
- ♦ Zestaw instrukcji umożliwiających wykonanie konkretnego zapytania do bazy,

- ◆ Zestaw instrukcji przetwarzających dane uzyskane w wyniku zapytania do bazy,
- ◆ Zestaw instrukcji czyszczących.

Co ważne, do przygotowania podstawowej wersji *Recordseta* nie jest potrzebne nie tylko ręczne wprowadzanie kodu PHP, ale nawet jego podgląd. Wszystkie operacje możemy wykonać w trybie graficznym za pomocą odpowiednich paneli dostępnych w programie Adobe Dreamweaver CS3.

## Panel Data — narzędzia budowy aplikacji

Sercem dynamicznych narzędzi Adobe Dreamweaver CS3 jest panel *Data* dostępny w górnym pasku lub menu (w zależności od konfiguracji programu) *Insert* (rysunek 4.11). W tym miejscu mamy dostęp do wszystkich istotnych elementów, potrzebnych do budowy dynamicznej aplikacji WWW. Przyjrzyjmy się tym narzędziom. Wstępnie wszystkie elementy panelu *Data* można podzielić na trzy fantastyczne kategorie:

- ◆ Narzędzia do pracy z zewnętrznymi danymi XML oraz tworzenia dynamicznych efektów w technologii Ajax.
- ◆ Narzędzia do współpracy z bazą danych.
- ◆ **Import Tabular Data** — osobne narzędzie do wprowadzania danych tabelarycznych, dostępne jako odrębna kategoria. Podczas importu plików zawierających odpowiednio sformatowane dane (np. pliku tekstowego zawierającego dane oddzielane przecinkiem, tabulatorem itp.) możemy łatwo i szybko zamieścić je w tabeli na stronie bez konieczności ręcznego wprowadzania danych.



**Rysunek 4.11.** Panel *Data* (w poprzednich wersjach programu nosił nazwę *Application*) zawiera szereg narzędzi ułatwiających budowę strony WWW opartej na bazie danych lub plikach XML

Narzędzia do pracy z zewnętrznymi danymi XML oraz tworzenia dynamicznych efektów w technologii Ajax:

- ◆ **Spry XML Data Set** — narzędzie pozwalające utworzyć nowe źródło danych technologii Spry (Ajax) na stronie WWW. W tym celu wykorzystywane są zewnętrzne pliki XML o odpowiedniej strukturze.
- ◆ **Spry Region** — narzędzie tworzenia obszaru dynamicznych danych budowanych na podstawie zewnętrznego źródła XML.
- ◆ **Spry Repeat** — narzędzie umożliwiające wyświetlanie kolejno wielu węzłów XML na pojedynczej stronie.
- ◆ **Spry Repeat list** — narzędzie umożliwiające tworzenie dynamicznej listy, menu na podstawie danych zawartych w zewnętrznym pliku XML.
- ◆ **Spry Table** — wygodne narzędzie do tworzenia tabeli danych na podstawie zewnętrznego pliku XML.

Narzędzia do współpracy z bazą danych:

- ♦ **Recordset** — narzędzie, które umożliwia połączenie strony z bazą, odczyt właściwych danych oraz ich prezentację na stronie.
- ♦ **Dynamic Data** — narzędzie umożliwiające wprowadzenie dynamicznego (pochodzącego z bazy danych) tekstu, listy lub pola w wybranym miejscu na stronie WWW.
- ♦ **Repeat Region** — narzędzie umożliwiające wyświetlanie wielu rekordów z bazy jednocześnie.
- ♦ **Show Region** — zestaw narzędzi umożliwiających wykonanie podstawowych instrukcji warunkowych na podstawie danych z bazy. Za pomocą narzędzi z tej sekcji możemy wyświetlać (lub nie) dane z bazy w zależności od ich dostępności. Instrukcja typu *pokaż jeśli jest* umożliwia dodanie najprostszej logiki do strony WWW.
- ♦ **Recordset Paging** — to zestaw narzędzi do zarządzania dużą ilością danych. Umożliwia łatwe dodanie elementów nawigacyjnych na kolejnych stronach zawierających jednocześnie wiele rekordów.
- ♦ **Display Record Count** — to także zestaw narzędzi do zarządzania dużą ilością danych. Umożliwia łatwe dodanie elementów nawigacyjnych na kolejnych stronach zawierających jednocześnie wiele rekordów.
- ♦ **Master Detail Page Set** — zestaw narzędzi do budowy stron typu *kategoria i produkt*. Strona *Master* zawiera listę wszystkich produktów danej kategorii, strona *Detail* — informacje o wybranym produkcie.
- ♦ **Insert Record** — fantastyczne narzędzia umożliwiające dodawanie rekordów do bazy danych.
- ♦ **Update Record** — narzędzia umożliwiające aktualizację rekordów bazy.
- ♦ **Delete Record** — narzędzia umożliwiające usuwanie wybranych rekordów z bazy.
- ♦ **User Authentication** — zestaw narzędzi ułatwiający budowę systemu uwierzytelniania i logowania użytkowników.
- ♦ **XSL Transformation** — zestaw narzędzi formatowania dokumentu XML lub danych RSS

Z punktu widzenia twórcy dynamicznej aplikacji WWW budowanej przy użyciu bazy danych i skryptów PHP, zestaw ten udostępnia nie tylko doskonałe narzędzia bazodanowe. Pozwala także na budowę elementów strony na bazie niezwykle popularnej i funkcjonalnej technologii Ajax. Tym zagadnieniom przyjrzymy się nieco uważniej w dalszej części książki.

Dzięki użyciu prostych narzędzi z kategorii *Data*, nauka programowania PHP oraz użycia bazy danych czy plików XML staje się po prostu przyjemnością.

## Przygotowanie źródła danych — Recordset

Do budowy dynamicznej listy kategorii na stronie *index.php* (strona główna serwisu) konieczne jest przygotowanie źródła danych — *Recordset*, które umożliwi połączenie strony z bazą, odczyt właściwych danych — wprowadzonych kategorii oraz ich prezentację na stronie. W tym celu

przechodzimy do zakładki (lub menu) *Data* dostępnej w górnym panelu *Insert*, po czym z dostępnych przycisków wybieramy polecenie (przycisk) *Recordset* (rysunek 4.12).



**Rysunek 4.12.** Przycisk *Recordset* umożliwia budowę nowego źródła danych. Na jednej stronie możemy wykorzystać kilka *Recordset*ów. Każdy z nich odpowiedzialny jest za inne działania



Wskazówka

Nowy *Recordset* możemy utworzyć także dwiema innymi metodami. Za pośrednictwem dostępnego w menu górnym *Window* panelu *Bindings* klikamy przycisk oznaczony symbolem +, a następnie z rozwijanej listy wybieramy polecenie *Recordset (Query)*. Możemy wykorzystać także menu górne *Insert*, a następnie przejść do pozycji *Data Objects* i z dostępnej w tym miejscu listy wybrać polecenie *Recordset*.

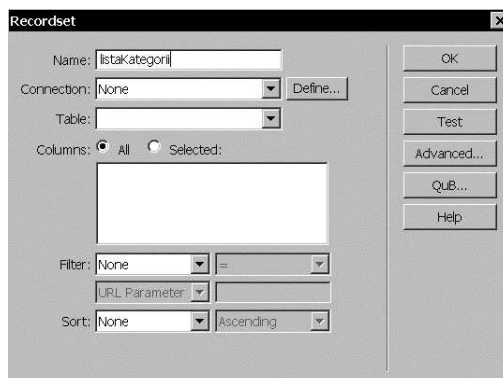
Niezależnie od wybranej metody, duże okno dialogowe *Recordset* pozwoli na precyzyjne ustalenie źródła danych dla naszej strony.

Panel *Recordset* dostępny jest w dwóch odmianach: *Basic* (*Podstawowy* — domyślnie aktywny), doskonały dla wszystkich rozpoczynających pracę z bazą danych, oraz *Advanced* (*Zaawansowany*), konieczny do wprowadzania bardziej zaawansowanych zapytań do bazy. W trybie *Basic* podstawowe elementy *Recordset*a dostępne są w postaci wizualnej i nie wymagają w zasadzie żadnej znajomości języka SQL. Podczas pracy w trybie *Advanced* niezbędne jest wprowadzanie kodu zapytania SQL ręcznie. Daje nam to duże możliwości dowolnego kształtowania samego zapytania, jak i relacji pomiędzy różnymi tabelami.

Do pierwszych prób połączenia szablonu z bazą danych wykorzystamy *Recordset* w trybie *Basic*. Jego funkcjonalność okazuje się zupełnie wystarczająca podczas tworzenia listy kategorii książek.

W polu *Name* wprowadzamy opisową nazwę zbioru danych. Warto pamiętać, by nie używać tu polskich znaków ani spacji. Nazwa powinna być dobrana tak, by nie kolidowała z nazwami tabel, pól bazy danych ani innymi elementami (nazwami) strony. W tym przypadku do budowy listy kategorii książek możemy wykorzystać prostą, a jednocześnie opisową nazwę typu — lista-kategorii (rysunek 4.13).

**Rysunek 4.13.** Okno *Recordset* umożliwia wprowadzenie na stronę zapytania SQL w trybie graficznym. To wspaniały sposób na poznanie i naukę pierwszych poleceń SQL dla użytkowników o mniejszym doświadczeniu



Pole *Connection* pozwala na wskazanie lub utworzenie właściwego pliku konfiguracyjnego umożliwiającego połączenie wszystkich stron serwisu z bazą danych. Pliki konfiguracyjne zapisywane są automatycznie w folderze *Connections* i zawierają wszystkie niezbędne dane dotyczące serwera baz danych, samej bazy, jej tabel, a także nazwę użytkownika oraz hasło. Użycie zewnętrznych plików konfiguracyjnych z rozszerzeniem \*.PHP w znaczący sposób zabezpiecza naszą stronę przed nieautoryzowanym dostępem do bazy. Dla jednego serwisu możemy przygotować kilka niezależnych sposobów łączenia z bazą (plików konfiguracyjnych), co pozwala ograniczyć dostęp do wybranych tabel bazy, podnosząc w ten sposób stopień ich zabezpieczenia.

Często podczas prac testowych wygodnie jest utworzyć dwa niezależne pliki konfiguracyjne. Jeden zawiera dane do serwera testowego *localhost* dostępnego na naszym dysku. Drugi zawiera dane niezbędne do współpracy z produkcyjną bazą danych na fizycznym serwerze WWW.

Podczas tworzenia pierwszego zbioru danych, *Recordset*, nie ma możliwości wyboru sposobu połączenia strony z bazą. Po prostu do tej pory nie utworzyliśmy jeszcze żadnego pliku konfiguracyjnego. W polu *Connection* pozostawiamy pozycję *None* i za pomocą przycisku *Define* przechodzimy do okna *Connections for Site siteRecenzje* (rysunek 4.14).



**Rysunek 4.14.** W celu nawiązania komunikacji z bazą danych trzeba przygotować plik konfiguracyjny — *Connection*. Dla wygody testowania naszej strony możemy przygotować dwa połączenia, lokalne do testowania strony na serwerze z naszego dysku oraz internetowe, niezbędne do poprawnego działania strony w Internecie

Jeśli wcześniej zdefiniowane były metody połączenia z bazą, ich lista widoczna będzie w wyświetlonym oknie. Aby zmienić ustawienia istniejącego pliku konfiguracyjnego, wybieramy go z listy, a następnie za pomocą przycisku *Edit* wchodzimy w tryb edycji jego właściwości.

W przypadku, gdy tworzymy pierwszy plik konfiguracyjny, na liście nie ma jeszcze widocznych żadnych połączeń. Za pomocą przycisku *New* przechodzimy do kolejnego okna — *MySQL Connection*. W tym miejscu uważnie wprowadzamy niezbędne dane konfiguracyjne.

- ♦ **Connection name (Nazwa połączenia)** — to opisowa nazwa połączenia, która powinna być krótka i zrozumiała. W przypadku użycia serwera testowego *localhost* możemy wprowadzić nazwę *localTest*. Jest ona krótka, zrozumiała i jednoznacznie wskazuje na użycie lokalnego serwera testowego. W przypadku pracy z serwerem internetowym nazwa w postaci *remoteTest* doskonale oddaje ten właśnie charakter połączenia.
- ♦ **MySQL Server (Serwer bazy danych MySQL)** — to adres serwera WWW. W przypadku pracy z serwerem Krasnal (lub innym zainstalowanym na naszym dysku) jego adres ma postać *localhost*.
- ♦ **User name (Nazwa użytkownika)** — to nazwa użytkownika, który ma dostęp do bazy danych. W przypadku użycia serwera Krasnal nazwa użytkownika o prawach administracyjnych (*Admin*) to *root*. W przypadku użycia innych serwerów konieczne jest sprawdzenie właściwej nazwy w ich dokumentacji.



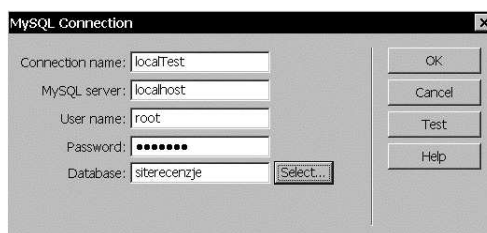
- ◆ **Password (Hasło)** — to hasło dostępu do bazy danych dla wybranego użytkownika. W przypadku pracy z serwerem Krasnal hasło domyślnego użytkownika to *krasnal*.
- ◆ **Database (Baza danych)** — to właściwa baza danych wykorzystywana podczas budowy danego serwisu WWW. Możemy oczywiście wprowadzić nazwę bazy ręcznie, jednak dużo wygodniej jest wskazać właściwą bazę z dostępnych dla wybranej lokalizacji (*localhost*) za pomocą przycisku *Select*. Do budowy strony z recenzjami książek wybieramy utworzoną wcześniej bazę — *siterecenzje*.

Przykładowe dane okna *MySQL Connection* dla serwera Krasnal przedstawione są poniżej (rysunek 4.15).

- ◆ *Connection name*: localTest
- ◆ *MySQL Server*: localhost
- ◆ *User Name*: root
- ◆ *Password*: krasnal
- ◆ *Database*: siteRecenzje

#### Rysunek 4.15.

Podczas tworzenia lokalnego połączenia użyjemy danych dotyczących bazy na naszym dysku. W przypadku połączenia internetowego w tym miejscu wprowadzamy dane umożliwiające połączenie i współpracę z bazą internetową



Za pomocą przycisku *OK* potwierdzamy wprowadzone właściwości połączenia i przechodzimy ponownie do okna *Connections for Site siteRecenzje*. Przycisk *Done* pozwala zachować ustawienia i przenosi nas z powrotem do okna edycji — *Recordset*.

Co bardzo istotne, bezpośrednio po zdefiniowaniu właściwości ustawień połączenia nie będzie ono automatycznie widoczne w polu *Connection*. Musimy ręcznie wskazać wprowadzoną nazwę *localTest*, by przejść do dalszej części panelu.

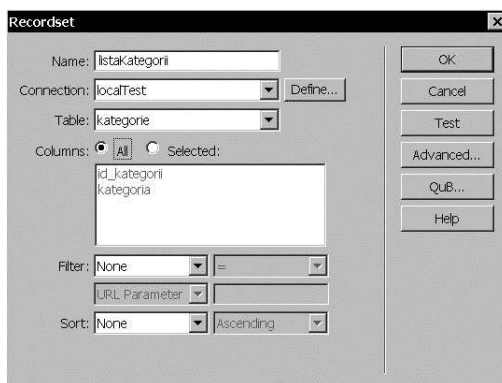
W polu *Table* wybieramy jedną z tabel w wybranej bazie danych. W przypadku bazy *siterecenzje* do budowy listy kategorii wybieramy tabelę *kategorie*, a następnie w polu *Columns* poniżej określamy, które pola (kolumny) tabeli chcielibyśmy wykorzystać.

W tym przypadku konieczne jest użycie wszystkich pól (*All*), jednak w wielu innych sytuacjach możemy wybrać opcję *Selected* (z wciśniętym klawiszem *Ctrl*) wskazać wybrane pola konieczne do budowy naszej strony.

Widoczne poniżej sekcje *Filter* oraz *Sort* pozwalają na wprowadzenie dodatkowych opcji dotyczących budowanego zbioru danych. W tej chwili nie ma konieczności ich modyfikacji (jednak zrobimy to nieco później), zatem w obu polach pozostawiamy domyślnie wybraną opcję — *None* (rysunek 4.16) i za pomocą przycisku *OK* potwierdzamy wprowadzone dane.

**Rysunek 4.16.**

*Podczas tworzenia Recordseta konieczne jest wskazanie odpowiedniego połączenia, tabeli z bazy oraz jej pól. W wielu przypadkach użyjemy wszystkich pól tabeli, jednak o ile jest to możliwe, starajmy się ograniczać ilość przesyłanych danych do minimum*



Każdy *Recordset* (źródło danych) widoczny jest automatycznie w panelu *Bindings* (panel *Bindings* otwiera się w chwili utworzenia nowego *Recordseta*). Po rozwinięciu zawartości za pomocą przycisku z symbolem + (widoczny tuż przed jego nazwą) wyświetla on listę wszystkich pól bazy danych dołączonych do bieżącego *Recordseta* podczas definiowania jego właściwości. W chwili użycia źródła danych generowanego na podstawie tabeli *kategorie* w panelu *Bindings* widoczne są dwie pozycje (rysunek 4.17):

- ♦ *id\_kategorii* — zawiera numeryczny identyfikator kategorii w tabeli
- ♦ *kategoria* — nazwa właściwej kategorii

**Rysunek 4.17.**

*W panelu Bindings widoczne są (w tym przypadku dwa) pola z wybranej tabeli bazy danych*



Do utworzenia dynamicznej listy menu wystarczy oczywiście użycie jedynie pola *kategoria*, które zawiera nazwy kategorii wprowadzone do bazy danych. Pole *id\_kategorii* wykorzystamy w nieco innym celu w dalszej części rozdziału.



Warto w tym momencie zwrócić uwagę na kod (przycisk *Source* widoczny w górnej części okna bieżącego dokumentu) HTML/PHP edytowanej strony. Dodanie dynamicznego źródła *Recordset* wprowadziło wiele istotnych zmian. Większość z nich widoczna jest w górnej części strony. Pomiędzy znacznikami kodu PHP — `<?php` tu treść kodu PHP na naszej stronie `?>` wprowadzonych jest kilka funkcji PHP ułatwiających połączenie z wybraną bazą, wyszukanie odpowiednich danych oraz wyświetlenie ich na stronie.

Podobnie jak w przypadku języka HTML, kod PHP zawarty jest w sekcji ograniczonej znacznikami. Zapis `<?php` oznacza początek kodu PHP na stronie, zaś odpowiednio znacznik `?>` jest jego zakończeniem. Pomiędzy znacznikami zawarte są właściwe funkcje i działania języka PHP/MySQL wykorzystane do wyświetlenia treści na podstawie utworzonego źródła danych.

Adobe Dreamweaver CS3 generuje bardzo ciekawy kod strony. Jest on połączeniem tradycyjnego HTML z elementami PHP umieszczonymi wewnątrz znaczników `<?php tu treść kodu PHP na naszej stronie ?>`. Co ważne, dzięki użyciu języka HTML możemy budować serwis w trybie graficznym, wprowadzając jedynie dynamiczne dane w wybranych sekcjach strony. Tego typu podejście zdecydowanie ułatwia pracę nawet zaawansowanym użytkownikom programu Adobe Dreamweaver CS3 o mniejszym doświadczeniu w programowaniu.

## Dynamiczne menu na stronie

Strona *index.php* po usunięciu nazw pozostałych kategorii w menu po lewej stronie zawiera teraz jedynie pojedynczą kategorię. Zaznaczamy ją, a następnie przechodzimy do palety *Bindings*. Tu wybieramy pole *kategoria* i za pomocą przycisku *Insert* widocznego w dolnej części panelu wprowadzamy dynamiczną zawartość na stronę (rysunek 4.18).

### Rysunek 4.18.

Za pomocą przycisku *Insert* możemy wprowadzać dynamiczne teksty na stronę. Łatwiej i szybciej jest je przeciągać z panelu *Bindings* w wybrane połączenie w oknie aktywnego dokumentu



Zaznaczona wcześniej statyczna pozycja menu zostaje w ten sposób automatycznie zastąpiona treścią w postaci `{listaKategori.kategoria}`. Zapis w tej właśnie postaci, a także inny kolor zaznaczenia informuje nas, że w tym miejscu wprowadzone zostały dynamiczne dane (rysunek 4.19). Pierwszy segment nazwy `{listaKategori.kategoria}` oznacza źródło danych — *Recordset*. W tym przypadku widoczna nazwa jednoznacznie wskazuje na *Recordset* `listaKategori`. Druga część nazwy widoczna bezpośrednio po kropce zawiera nazwę pola (`kategoria`) wykorzystanego do budowy menu na stronie.

Niezwykle wygodną funkcją Adobe Dreamweavera CS3 jest możliwość podglądu dynamicznej zawartości z bazy danych na bieżącej stronie bez konieczności uruchamiania przeglądarki internetowej. W tym celu należy zlokalizować przycisk *Live Data View* widoczny w górnej części okna aktywnego dokumentu. Po uruchomieniu podglądu na liście kategorii powinna ukazać się jedynie pierwsza pozycja z bazy (rysunek 4.20).

W niektórych sytuacjach użycie podglądu *Live Data View* nie daje dobrych rezultatów. Dotyczy to szczególnie sytuacji, gdy domyślny kolor podświetlenia dynamicznych danych pokrywa się z kolorystyką lub tłem tekstu. Możemy to łatwo zmienić za pomocą odpowiedniej edycji preferencji ogólnych programu. Za pomocą polecenia *Preferences* (*Ctrl+U*) dostępnego w menu



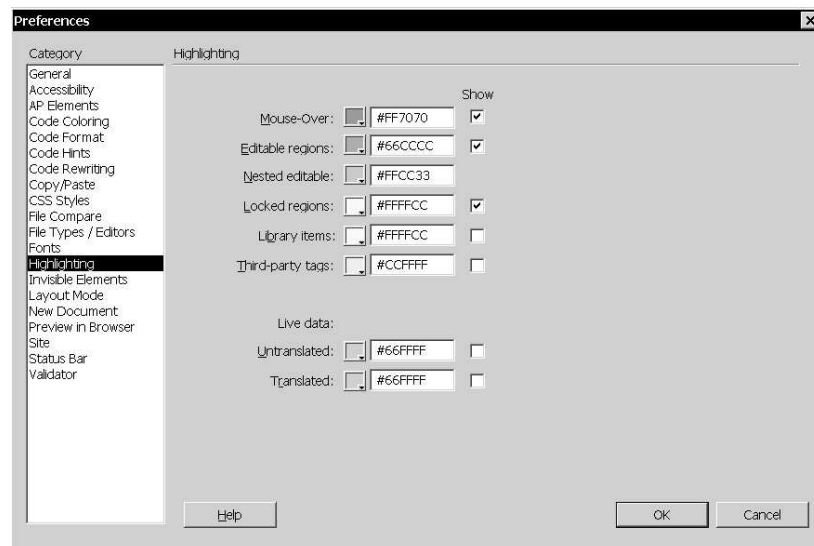
**Rysunek 4.19.** Dynamiczna nazwa kategorii na stronie. W domyślnej postaci dynamiczne pole tekstowe widoczne jest w jasnoniebieskim podświetleniu. W wielu przypadkach skutecznie utrudnia to pracę.



**Rysunek 4.20.** Za pomocą przycisku Live Data View możemy podglądać dynamiczne treści bezpośrednio na stronie WWW. Znacznie ułatwia to późniejsze formatowanie i planowanie rozmieszczenia elementów na stronie

górnym *Edit* przechodzimy do sekcji *Highlighting* (rysunek 4.21). Jeśli w dolnej sekcji *Live Data* w obu przypadkach ustawimy brak koloru, Adobe Dreamweaver CS3 wyświetli dane w oryginalnej ostylowanej postaci. Tego typu podgląd bardzo ułatwia dalszą pracę (rysunek 4.22).

**Rysunek 4.21.** Wokół Preferencji, w kategorii *Highlighting* mamy możliwość modyfikacji koloru podświetlenia dynamicznych danych na stronie. W wielu przypadkach usunięcie podświetlenia jest bardzo wygodnym rozwiązaniem





**Rysunek 4.22.** Dynamiczne dane bez podświetlenia prezentują się dużo lepiej. Taki tryb podglądu jest bardzo przydatny podczas edycji stylów CSS i formatowania elementów strony

Prawdopodobnie dla wielu Czytelników widoczny podgląd jest pewnym zaskoczeniem. Skoro nasz *Recordset* miał umożliwić wyświetlanie wszystkich kategorii książek, czemu widoczny jest tylko pierwszy rekord bazy danych? Nie ma tu nic dziwnego. Do tej pory nie podjęliśmy żadnych działań umożliwiających wyświetlenie wszystkich pozycji z pola kategoria. Spróbujmy to zmienić.



Automatycznie podczas użycia dowolnego źródła danych (*Recordset*) mamy dostęp jedynie do pierwszego rekordu bazy. Jego treść widoczna jest w trybie *Live Data View*, a także na podglądzie strony w przeglądarce internetowej — polecenie *Preview In Browser (F12)* dostępne w menu górnym *File*.

W celu wyświetlenia większej liczby rekordów na stronie trzeba wprowadzić dodatkowe funkcje języka PHP. Oczywiście w pierwszej chwili brzmi to raczej mało zachęcająco, jednak nie jest to wcale trudne zadanie. Musimy jedynie precyzyjnie zaznaczyć pierwszą pozycję menu w lewej kolumnie, zawartą pomiędzy znacznikami `<li></li>` (rysunek 4.23), a następnie w górnej sekcji strony zlokalizować i wykorzystać przycisk *Repeat Region* dostępny w zakładce (lub menu) *Data* (rysunek 4.24).



**Rysunek 4.23.** W dolnej części strony widoczna jest hierarchia znaczników HTML naszej strony. Ostatni znacznik na liście przyporządkowany jest do obiektu aktywnego na stronie. Widoczne tu selektory możemy wykorzystywać także do precyzyjnego zaznaczania wybranych elementów strony



**Rysunek 4.24.** Za pomocą przycisku *Repeat Region* możemy wprowadzać cykliczne, powtarzalne elementy strony na podstawie treści pobranych z bazy danych. Obszar *Repeat Region* budowany na podstawie znaczników pozycji listy `<li>` pozwala na budowę dynamicznego menu na stronie



Użycie polecenia *Repeat Region* pozwala automatycznie utworzyć funkcję języka PHP, która wyświetla dowolną liczbę rekordów z bazy danych.

W oknie *Repeat Region* znajdują się trzy nieskomplikowane opcje. W polu *Recordset* upewniamy się, że widoczna jest nazwa właściwego źródła danych — lista kategorii. Sekcja *Show* pozwala na wybór ilości rekordów jednorazowo prezentowanych na stronie (rysunek 4.25).



**Rysunek 4.25.** W oknie konfiguracyjnym *Repeat Region* wprowadzamy docelową liczbę rekordów (elementów z bazy danych), jakie chcielibyśmy wyświetlić na stronie. Nie zawsze użycie opcji *All* daje dobre rezultaty

- ♦ *Show... Records at a Time* — to wybór dowolnej, skończonej liczby rekordów widocznych na stronie
- ♦ *Show All records* wyświetla pełną listę rekordów, niezależnie od jej ilości. W sytuacji, gdy lista rekordów jest bardzo długa, użycie tej opcji nie tylko spowalnia proces wyświetlania strony, ale także nie wygląda później zbyt elegancko.

W przypadku budowy listy kategorii książek wybierzemy oczywiście opcję *All Records* (nie mamy wielu kategorii), co umożliwi wyświetlenie wszystkich wprowadzonych kategorii w obszarze menu. W ten sposób użytkownicy serwisu będą mogli łatwo i szybko wybrać właściwą kategorię, by wyszukać odpowiednią recenzję ulubionej książki.

Dzięki użyciu podglądu *Live Data View* menu widoczne w lewej kolumnie zawiera teraz listę wszystkich kategorii wprowadzonych do bazy danych. Nie było to chyba trudne zadanie (rysunek 4.26).



**Rysunek 4.26.** Dzięki użyciu polecenia *Repeat Region* dynamiczna lista kategorii na podstawie tabeli kategorii z bazy danych znalazła się na naszej stronie

## Łącza tymczasowe

Aby menu nawigacyjne kategorii książek umożliwiło przejście do innej strony serwisu, konieczne jest dodanie mu odpowiednich łączy (linków). Na tym etapie budowy całego serwisu trudno jest jeszcze planować właściwe połączenia pomiędzy stronami. W tej chwili nie mamy jeszcze żadnych stron, do których miałyby one prowadzić. Oczywiście moglibyśmy pozostawić menu w bieżącej postaci, jednak wprowadzenie choćby tymczasowych linków znacznie ułatwi nam późniejsze działania, a i teraz wprowadzi nieco zmian w wyglądzie bieżącej strony.

Menu kategorii książek generowane jest dynamicznie na podstawie treści pobranych z bazy danych. W związku z tym na stronie *index.php* znajduje się jedynie dynamiczne odwołanie do wybranego pola `{listaKategorii.kategoria}` tabeli. Dzięki użyciu niezwykle wygodnej funkcji *Repeat Region*, wyświetlamy w tym miejscu pełną listę kategorii. Oznacza to, nie mamy możliwości dodania łączy ręcznie do każdej pozycji menu. Jak łatwo się domyślić, muszą być one także dynamicznie generowane.

W tej chwili wprowadzimy jednak jedynie proste linki zastępcze. W tym celu możemy wykorzystać znacznik `#`. Wprowadzenie łączy w takiej postaci oczywiście nie spowoduje poprawnego działania strony, jednak w znaczący sposób ułatwi późniejszą modyfikację.

Zaznaczamy na stronie dynamiczny tekst w postaci `{listaKategorii.kategoria}`, a następnie w polu *Link* wprowadzamy znacznik `#`. Potwierdzamy to za pomocą klawisza *Enter* (rysunek 4.27).



**Rysunek 4.27.** Za pomocą selektora `<li>` na liście w dolnej części strony możemy precyzyjnie zaznaczyć wybrany element strony



Dodawanie łączy (linków) należy zawsze potwierdzać za pomocą klawisza *Enter*!

Niestety, dodanie zastępczego odnośnika wprowadziło niekorzystną zmianę na naszej stronie. Do tej pory atrakcyjnie wyglądające (na podglądzie lub przy użyciu funkcji *Live Data View*) menu stało się niebieskie i podkreślone. Nie jest to nic niezwykłego. W ten sposób przeglądarki domyślnie wyświetlają łączy na każdej stronie. Dopóki za pomocą definicji CSS nie zmienimy wyglądu odnośnika, dopóty przyjmuje on standardową postać — niebieski podkreślony tekst. Co ciekawe, wielkość oraz styl odnośnika pozostały bez zmian, zmienił się jedynie kolor i dodane zostało podkreślenie (rysunek 4.28).



**Rysunek 4.28.** Po dodaniu łącza wygląd pozycji menu zmienił się. W tej chwili elementy listy przybrały domyślny wygląd odnośników na stronie. Są koloru niebieskiego i mają dodatkowe podkreślenie

## Dodatkowe style CSS

By nadać elementom menu wcześniej określony wygląd, konieczne jest przygotowanie kolejnych definicji stylów kaskadowych CSS. Nie jest to szczególnie trudne zadanie, jednak w tym miejscu postaramy się dodać liście menu kilka ciekawych możliwości.

W dzisiejszych czasach, w dobie burzliwego rozwoju Internetu i nowoczesnych technologii, trudno sobie wyobrazić, by przyciski ekranowe (np. menu z listą kategorii) pozostawały statyczne na stronie. Zazwyczaj nadajemy im typowy efekt *RollOver*, czyli zmianę wyglądu po najechaniu i naciśnięciu kursorem myszki. Tego typu efekty często kojarzone są z budowaniem specjalnych elementów graficznych, które następnie są podmieniane za pomocą skryptów języka Java Script w chwili najechania na nie kursorem myszy. Na szczęście takie podejście dziś już może odejść w niepamięć. Dzięki użyciu stylów kaskadowych możemy przygotować niemal identyczny wizualnie efekt bez konieczności użycia grafiki i Java Scriptu.

Zanim przejdziemy do budowy stylów, przypomnijmy sobie strukturę menu kategorii. Całość znajduje się w sekcji `<div#sidebar1>`, a następnie wewnątrz sekcji listy nienumerowanej, czyli `<ul></ul>`. Jedyna pozycja listy `{listaKategorii.kategoria}` zamknięta jest wewnątrz znacznika `<li></li>`. W wyniku dodania odnośnika w postaci `#` pole `{listaKategorii.kategoria}` zawiera jeszcze dodatkowy tag języka HTML — `<a href="#">`. Jest to znacznik łącza (linku) tymczasowego (`#`), który w tej chwili nie powoduje żadnych konkretnych działań na stronie.

A analizując całą sekcję menu na stronie, nie znajdziemy w tej chwili zbyt wiele kodu. Mamy jedynie pojedyncze dynamiczne pole `{listaKategorii.kategoria}` wraz z dodanym odnośnikiem `<a>`. Pełna zawartość listy jest dynamicznie pobierana z bazy i za pomocą obszaru *Repeat Region* wielokrotnie powielana na stronie. W tej sytuacji nasze działania związane z formatowaniem menu dotyczą jedynie dynamicznego pola `{listaKategorii.kategoria}` (rysunek 4.29).

Krokiem, który spowodował utratę oryginalnego wyglądu strony, było dodanie łącza do dynamicznego pola tekstowego. Większość przeglądarek przedstawia odnośniki (linki), a więc elementy zawarte wewnątrz znacznika `<a>`, w swój domyślny sposób (niebieskie teksty z podkreśleniem). Chcąc zmienić ten stan, musimy pokusić się o modyfikację wyglądu znacznika `<a>`.

Niestety, nie możemy zrobić tego globalnie. Redefinicja wyglądu łącza na stronie spowoduje zmianę wyglądu wszystkich linków. O ile w przypadku menu kategorii pewien ustalony wygląd sprawdzi się doskonale, o tyle w głównej sekcji dokumentu może wyglądać bardzo źle. Oznacza to, że musimy w jakiś sposób odróżnić definicję wyglądu znacznika `<a>` sekcji `<div#sidebar1>` od pozostałych. W zasadzie rozwiązanie widoczne jest już w poprzednim zdaniu. Jeśli wprowadzimy redefinicję znacznika jedynie w sekcji `<div#sidebar1>`, pozostałe odnośniki pozostaną bez zmian, co było naszym założeniem. Zatem do dzieła.



```

67 </li>Kontakt </li>
68 </ul>
69 <!-- end #header --></div>
70 <!-- InstanceBeginEditable name="menuLeft" -->
71 <div id="sidebar1">
72 <h3>Kategorie</h3>
73 <ul>
74 <?php do { ?>
75 <li><a href="#"><?php echo $row listaKategorii['kategoria']; ?></a></li>
76 <?php } while ($row listaKategorii = mysql_fetch_assoc($listaKategorii)); ?></ul>
77 <!-- end #sidebar1 --></div>
78 <!-- InstanceEndEditable --><!-- InstanceBeginEditable name="mainArea" -->
79 <div id="mainContent">
80 <h1>BookShelf Online - witamy na naszych stronach</h1>
81 <p></p>
82 <h2>To jest nagłówek drugiego stopnia czyli H2. </h2>
83 <p>Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing.
Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio. Donec et ipsum
et sapien vehicula nonummy. Suspendisse potenti. Fusce varius urna id quam. Sed neque mi, varius eget,
tincidunt nec, suscipit id, libero. In eget purus. </p>

```

**Rysunek 4.29.** Dzięki użyciu bazy danych cała lista kategorii wraz z linkami zapisana jest w niewielkim fragmencie kodu

Klikamy gdziekolwiek dynamiczne pole {listaKategorii.kategoria} (w ten sposób możemy szybko zaznaczyć sekcję znacznika <a>), a następnie w dolnej części strony wskazujemy ostatni na liście hierarchii elementów strony znacznik <a> (rysunek 4.30). Otwieramy paletę *CSS Styles* (*Shift F 11*) i dodajemy nowy styl.



**Rysunek 4.30.** Za pomocą selektorów w dolnej części strony zaznaczamy całe łącze — znacznik <a>

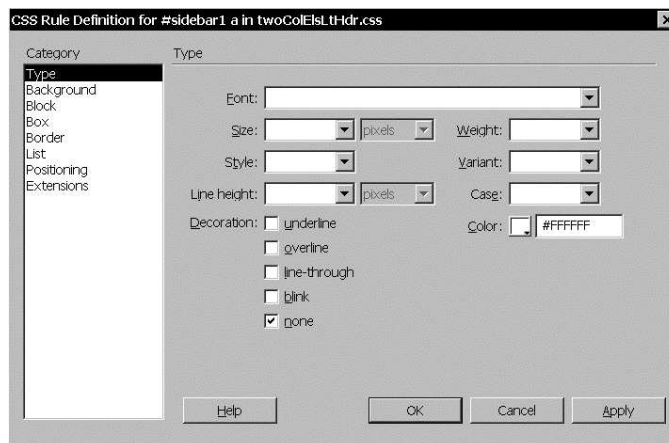
Adobe Dreamweaver CS3 automatycznie zaproponuje styl typu *Advanced*, nadając mu nazwę selektora w postaci `.twoColElsLthdr #container #sidebar1` a. Podobnie jak robiliśmy to wcześniej, możemy nieco ją skrócić, doprowadzając ostatecznie do postaci `#sidebar1` a. Zapis ten oznacza: wszystkie odnośniki (znacznik <a>), które znajdują się w sekcji `#sidebar1`, będą sformatowane na podstawie definicji, jakie niebawem wprowadzimy.

W oknie *CSS Rule Definition* modyfikujemy jedynie dwie właściwości stylu (rysunek 4.31):

- ◆ *Color* — doprowadzamy go do oryginalnej postaci widocznej przed dodaniem łącza.
- ◆ *Decoration none* — w ten sposób usuwamy (jeśli potrzeba) zbyteczne podkreślenie elementów menu kategorii.

**Rysunek 4.31.**

Okno CSS Rule  
Definicja zawiera  
graficzny interfejs  
budowy i edycji stylów  
CSS. W tym miejscu  
zmienimy jedynie kolor  
odnośnika oraz  
usuniemy zbyteczne  
podkreślenie



Za pomocą przycisku *Apply* możemy od razu sprawdzić działanie stylu na stronie. Jeśli wszystko wygląda zgodnie z naszymi oczekiwaniami, za pomocą przycisku *OK* kończymy budowę stylu. W tym momencie nie ma potrzeby dokonywać tu jeszcze więcej modyfikacji. Dodaniem innych efektów zajmiemy się dużo później.

## Wykorzystanie obiektów Library

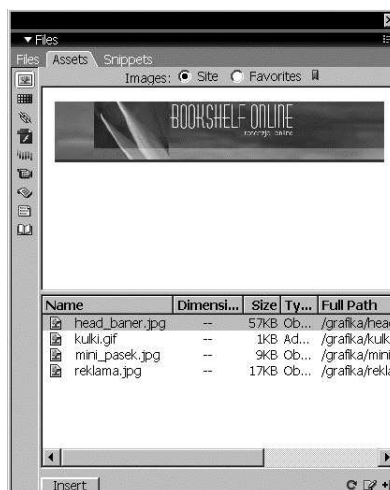
Adobe Dreamweaver CS3 posiada jeszcze jedno wspaniałe narzędzie, które doskonale sprawdzi się podczas naszej dalszej pracy. Obiekty *Library*, bo o nich mowa, to dowolne elementy strony, fragmenty kodu czy wręcz całe funkcjonalne moduły zapisane w bibliotece (*Library*) naszego *site'a*. Ogromną ich zaletą jest możliwość wykorzystania w stałej, niezmiennej i wręcz zablokowanej do edycji postaci na wielu różnych stronach naszego serwisu. Jednak największą korzyścią z obiektów *Library* mamy w chwili ich edycji. Zmieniając obiekt zapisany w bibliotece (*Library*), automatycznie aktualizujemy ten właśnie fragment kodu na wszystkich stronach, które go wykorzystują. W pewnym sensie przypomina to działanie szablonu, ponieważ podobnie jak szablon, on także jest zablokowany i niedostępny do edycji na kolejnych stronach. Podczas typowej pracy wykorzystujemy jedynie stały, zapisany w bibliotece obiekt na wielu stronach serwisu. W odpowiednim momencie modyfikujemy jego zawartość, co spowoduje aktualizację wszystkich wykorzystujących go stron. Brzmi to fantastycznie i dokładnie tak działa.

W przypadku strony *index.php* zbudowaliśmy właśnie menu nawigacyjne zawierające listę kategorii książek, których recenzje można znaleźć w naszym serwisie. Nie jest to jednak w pełni funkcjonalne menu. Do jego pełnego działania potrzebujemy dodać jeszcze łącza (linki), które umożliwią przeglądanie książek wybranej kategorii. W chwili obecnej nie mamy jeszcze przygotowanych takich właśnie stron docelowych. Znacznie utrudnia to przygotowanie właściwych łączy (linków). W tej sytuacji doskonałym rozwiązaniem okaże się zastosowanie obiektu *Library*, który w oryginalnej postaci wykorzystamy do budowy kolejnych stron serwisu. Gdy powstaną już strony umożliwiające przeglądanie książek wybranej kategorii, dodamy odpowiednie łącza dla obiektu z biblioteki. Dzięki temu wszystkie strony, które wykorzystują taki obiekt, jednocześnie zostaną zaktualizowane. To nie tylko bardzo wygodne rozwiązanie. Pozwala znacznie przyspieszyć dalszą pracę, skupiając się na jej istotnych elementach. Drobne modyfikacje typu ostatni szlif faktycznie można wprowadzić nieco później.

Obiekty *Library* dostępne są jedynie podczas pracy z całą witryną. Nie ma możliwości ich użycia w sytuacji, gdy budujemy jedynie pojedynczą stronę i nie przygotowaliśmy pełnej definicji *site'a* (katalogu). W celu użycia tych obiektów przechodzimy do palety *Assets (F11)* dostępnej w menu górnym Window (rysunek 4.32).

#### Rysunek 4.32.

*Paleta Assets przechowuje cenne zasoby naszego serwisu. Aby skorzystać z jej usług, należy wcześniej zdefiniować site. Jedynie w tym trybie mamy dostęp do palety Assets, a w niej do szablonów, obiektów biblioteki czy kolorów*



Podobnie jak podczas pracy z bazą danych, wszystkie przydatne palety zgrupowane są w pojedynczym oknie i nie ma konieczności wywoływania ich z menu górnego *Window*. Klikając kolejne zakładki panelu *Files* (a także *Application* w przypadku pracy z dynamicznymi danymi), możemy szybko przechodzić pomiędzy odpowiednimi panelami. Panel *Assets* dostępny jest w tej samej sekcji co podstawowe okno *Files*, standardowo widoczne po prawej stronie ekranu.

Paleta *Assets* zawiera szereg niezwykle przydatnych zasobów, które z powodzeniem możemy wykorzystywać podczas pracy z serwisem. Dostępne w pionowej kolumnie po lewej stronie przyciski pozwalają na szybkie przełączenie podglądu zasobów różnego typu.

- ◆ **Images** — zawiera listę i podglądy wszystkich plików graficznych dostępnych w katalogu głównym naszego *site'a*. Jeśli nieco rozszerzymy podgląd zasobów palety *Assets*, widoczne staną się także dodatkowe, całkiem przydatne informacje. Pośród nich znajdziemy oczywiście nazwę oraz: *Dimensions*, czyli rozmiary grafiki, *Size* — jej wagę, *Type* — typ pliku graficznego oraz *Full Path* — pełną ścieżkę dostępu do wybranej grafiki. Podgląd oraz zestaw tych informacji okazują się bardzo przydatne podczas pracy z większymi serwisami, gdzie nie sposób zapamiętać wszelkich informacji o każdym użytym pliku graficznym.
- ◆ **Colors** — to pełna lista wszystkich kolorów wykorzystanych podczas budowy naszej witryny. Oczywiście dotyczy ona przede wszystkim tych kolorów, jakie wykorzystaliśmy do budowy stylów kaskadowych. Jeśli później podczas budowy kolejnych elementów strony czy stylów chcielibyśmy zachować zgodność kolorystyczną na stronie, możemy podczas ustalania ich barwy wykorzystać tę właśnie listę i wskazać jeden z użytych już wcześniej kolorów. Na podglądzie w górnej części okna widoczny jest zarówno zapis koloru w kodzie szesnastkowym, jak i jego definicja na podstawie RGB. Może to być niezwykle przydatne szczególnie podczas tworzenia dodatkowych elementów graficznych, gdy potrzebujemy dopasować ich kolor do elementów strony.

- ♦ **Links** — przedstawia pełną listę wszystkich zewnętrznych adresów URL wykorzystanych podczas budowy naszej strony. Jeśli pewien odnośnik do zewnętrznej strony czy adresu e-mail chcielibyśmy wykorzystać w wielu miejscach, wystarczy zaznaczyć odpowiedni tekst lub grafikę i przeciągnąć właściwy link z palety *Assets* na zaznaczony element na stronie.
- ♦ **Flash** — to naturalnie zestawienie wszystkich elementów Flash (SWF) dostępnych w katalogu głównym *site'a*.
- ♦ **Shockwave** — dziś już mało popularna, lista wszystkich elementów Shockwave (DCR) dostępnych do użycia na naszej stronie. Pliki typu Shockwave budowane są za pomocą fantastycznego, choć nieco zapomnianego programu Macromedia (Adobe) Director.
- ♦ **Movies** — to pełna lista filmów wideo zapisanych w głównym katalogu naszej witryny.
- ♦ **Scripts** — to lista wszystkich zewnętrznych skryptów (np. Java Script) wykorzystanych do budowy bieżącego serwisu.
- ♦ **Templates** — zawiera pełną listę wszystkich szablonów stron WWW aktywnego serwisu.
- ♦ **Library** — to biblioteka wspólnych elementów, jakie w stałej postaci możemy wykorzystywać na naszej stronie. Dowolna modyfikacja elementu z biblioteki automatycznie aktualizuje wszystkie strony, jakie z niego korzystają.

Nas oczywiście najbardziej interesuje możliwość praktycznego wykorzystania biblioteki (*Library*). Jest to niezwykle proste. Musimy precyzyjnie zaznaczyć fragment strony, jaki chcielibyśmy zapisać w bibliotece, i przeciągnąć go do palety *Assets*. W niektórych sytuacjach trudno jest przeciągnąć zaznaczony element i w takiej chwili możemy wykorzystać także przycisk *New Library Item* widoczny w dolnej części palety *Assets*. W ten sposób unikamy kłopotliwego niekiedy przeciągania obiektów i precyzyjnie umieszczamy właściwy element w bibliotece.

W przypadku budowy menu z listą kategorii musimy dokładnie zaznaczyć obszar, jaki chcielibyśmy zamieścić w bibliotece. W tym celu klikamy widoczną dynamiczną pozycję listy, a następnie przechodzimy do widoku kodu. Wprawdzie możliwe jest zaznaczanie także w trybie *Design*, jednak dużo łatwiej w ten sposób o pomyłkę. Pracując w widoku kodu strony, precyzyjnie zaznaczamy jedynie te fragmenty, które są faktycznie konieczne.

Dzięki zaznaczeniu fragment kodu z listą kategorii jest właśnie wyróżniony na ekranie. Lokalizujemy sekcję znaczników `<ul></ul>` i zaznaczamy cały fragment kodu tej listy. Należy zwrócić baczną uwagę, by zaznaczony został zarówno początek listy `<ul>`, cała zawartość wraz z kodem PHP, jak i znacznik końcowy listy `</ul>` (rysunek 4.33).

Teraz za pomocą polecenia *New Library Item* dostępnego w menu bocznym palety (rysunek 4.34) dodajemy całą listę kategorii do biblioteki. W celu łatwiejszego odnalezienia właściwego obiektu w bibliotece wprowadzamy jej opisową nazwę — `listaKategorii` (rysunek 4.35). Po zatwierdzeniu nazwy otrzymamy komunikat o aktualizacji łączy, który naturalnie potwierdzamy.

W rezultacie tych działań menu z listą kategorii zostało wyróżnione kolorem oraz zablokowane do bezpośredniej edycji. W celu modyfikacji elementów menu konieczna jest teraz edycja oryginalnego fragmentu kodu z biblioteki.

```

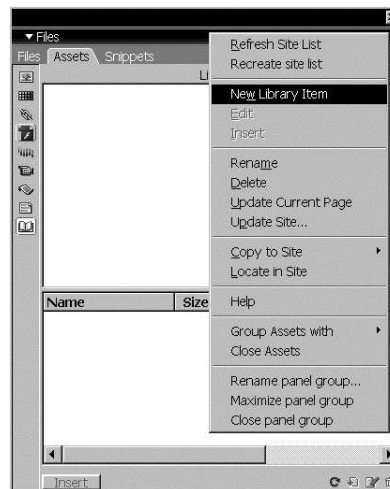
67 <li>Kontakt </li>
68 </ul>
69 <!-- end #header --></div>
70 <!-- InstanceBeginEditable name="menuLeft" -->
71 <div id="sidebar1">
72 <h3>Kategorie</h3>
73 <ul>
74 <?php do { ?>
75 <li><a href="#"><?php echo $row_listaKategorii['kategoria']; ?></a></li>
76 <?php } while ($row_listaKategorii = mysql_fetch_assoc($listaKategorii)); ?></ul>
77 <!-- end #sidebar1 --></div>
78 <!-- InstanceEndEditable --><!-- InstanceBeginEditable name="mainArea" -->
79 <div id="mainContent">
80 <h1>BookShelf Online - witamy na naszych stronach</h1>
81 <p></p>
82 <h2>To jest nagłówek drugiego stopnia czyli H2. </h2>
83 <p>Quisque ornare risus quis ligula. Phasellus tristique purus a augue condimentum adipiscing.
Aenean sagittis. Etiam leo pede, rhoncus venenatis, tristique in, vulputate at, odio. Donec et ipsum
et sapien vehicula nonummy. Suspendisse potenti. Fusce varius urna id quam. Sed neque mi, varius eget,
tincidunt nec, suscipit id, libero. In eget purus. </p>
84 <h3>H3 oto nagłówek trzeciego stopnia</h3>
85 <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent aliquam, justo convallis
luctus rutrum, erat nulla fermentum diam, at nonummy quam ante ac quam. Maecenas urna purus, fermentum
id, molestie in, commodo porttitor, felis. Nam blandit quam ut lacus. Quisque ornare risus quis
ligula. Phasellus tristique purus a augue condimentum adipiscing. Aenean sagittis. Etiam leo pede,
rhoncus venenatis, tristique in, vulputate at, odio.
86 <!-- end #mainContent -->
87 </div>

```

**Rysunek 4.33.** Zaznaczony fragment listy zamknięty wewnątrz znaczników `<ul>`. Możemy zaznaczać elementy bezpośrednio na stronie lub skorzystać z selektorów widocznych w jej dolnej części

#### Rysunek 4.34.

Za pomocą przycisku *New Library Item* tworzymy nowy obiekt w bibliotece. Obiekty te możemy tworzyć także poprzez przeciągnięcie aktywnej treści strony w okno *Library*



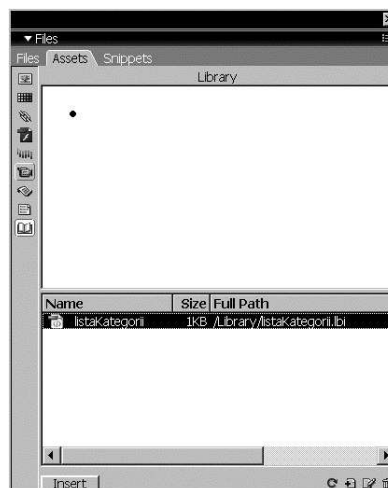
Jeśli teraz na podstawie strony *index.php* powstaną nowe strony, każda z nich wykorzysta wprowadzony obiekt z biblioteki. Jego edycja uaktualni wszystkie strony naszego serwisu. A to właśnie było naszym założeniem.

## Dynamiczna zawartość strony startowej

Kolejne zmiany dotyczą już głównej części strony *index.php* — `<div#mainContent>`. Tu w miejsce widocznych właśnie statycznych informacji tekstowych wprowadzimy treść zapisaną w tabeli *informacje* naszej bazy danych. Ponadto poniżej dodamy kolejny moduł serwisu — *Nowości*/

**Rysunek 4.35.**

Nowy obiekt *listaKategorii* został zapisany w bibliotece. Od tej chwili możemy wykorzystywać go wielokrotnie w naszej pracy. Aktualizacja obiektu z biblioteki automatycznie modyfikuje wszystkie strony serwisu, jakie go wykorzystują



*Aktualności.* Znajdą się tu najnowsze informacje dotyczące działania serwisu, opisy, aktualności oraz komunikaty administracyjne zapisane w bazie danych. A by zanadto nie zaśmiecać strony głównej nadmiarem różnorodnych danych i informacji, umieścimy tu jedynie 2 – 3 najnowsze rekordy z bazy. Lista wszystkich aktualności dostępna będzie na stronie *Archiwum nowości*.

Do pobrania ogólnych informacji o firmie z bazy danych konieczne jest utworzenie kolejnego źródła danych zwanego *Recordset*. W ten sposób możliwe będzie nawiązanie połączenia z zupełnie inną tabelą, wyszukanie właściwych informacji oraz umieszczenie ich w obszarze edytowalnym `<div#mainContent>` strony.

Nowy *Recordset* utworzymy tym razem za pomocą palety *Bindings* (*Ctrl+F10*) dostępnej w menu górnym *Window*. Za pomocą niewielkiego przycisku z symbolem `+` dodajemy nowy *Recordset* (rysunek 4.36) i przechodzimy w tryb edycji jego właściwości. W polu *Name* wprowadzamy nazwę `mainInfo`, na liście *Connection* wybieramy utworzony wcześniej plik konfiguracyjny — *localTest*, tabelę *informacje*, a następnie w sekcji *Columns* wybieramy opcję *All*, która umożliwi nam dostęp do wszystkich pól tabeli.

Podobnie jak w przypadku listy kategorii, w tym momencie nie zmieniamy domyślnych ustawień (*None*) w sekcji *Filter* oraz *Sort* (rysunek 4.37).

Nowy *Recordset* widoczny jest w panelu *Bindings*. Po rozwinięciu jego zawartości widoczne stają się wszystkie pola utworzonej wcześniej tabeli — *informacje*. Dzięki odpowiedniemu zaprojektowaniu pól tabeli, wprowadzenie dynamicznej treści na stronę tytułową nie sprawi nam żadnych problemów.

Przypomnijmy sobie bieżącą strukturę logiczną strony *index.php*. Jej tytuł zawarty jest wewnątrz znaczników `<h1></h1>`. Poniżej znajduje się niewielki tekst wprowadzający — nagłówek zawarty wewnątrz znaczników `<h2></h2>`. Właściwa treść tej strony zapisana jest w kolejnych akapitach (znaczniki `<p></p>`) tekstu. Jeśli podczas wprowadzania statycznej treści strony zdecydowaliśmy się na użycie dodatkowej grafiki czy fotografii, widzimy ją poniżej nagłówka, a jednocześnie ponad właściwym tekstem strony.