



Appendix A

USING LANGUAGES OTHER THAN ENGLISH IN MYSQL

MySQL is a truly international database. A standard installation has support for more than 20 different encoding systems, including UTF-8 (since version 4.1). This means that MySQL now supports just about every living language in the world—and some dead ones, too. If you want to store data in Hmong, Bugis, or Tamazight, you're out of luck: UTF-8 doesn't support them *yet*—but it will eventually.

How MySQL uses character sets and collation

It's important to realize that the default character set of a MySQL server has no bearing whatsoever on the range of languages in which data can be stored. The choice of character set affects just two things:

- The characters permitted in database, table, and column names
- The default sort order of query results

Collation also determines the sort order, but it offers a greater degree of control, and it is the setting that concerns you most in a multilingual or non-English environment. Until the introduction of collation in MySQL 4.1, the only way to change the sort order of results was to change the character set for the entire server. This was extremely inconvenient for storing data in more than one language because Spanish, English, and German, for instance, all follow different sorting rules.

MySQL's default collation is `latin1_swedish_ci`, which reflects the fact that MySQL AB is based in Sweden. Fortunately, the sort orders for Swedish and English are the same. So, unless you create databases in other languages, or need to use accented characters, you can forget about collation.

Table A-1 shows the available settings for Western European languages using the `latin1` character set. Table A-2 shows the different ways some of the main ones sort the same items. As you can see, the default Swedish order sorts *ü* after *x*, so for English-based material that uses accents, `latin1_general_ci` is the preferred setting.

Table A-1. Collation settings in MySQL for the `latin1` character set

Collation	Description
<code>latin1_bin</code>	Binary, follows the <code>latin1</code> encoding order (all uppercase letters come before lowercase ones: <i>Z</i> comes before <i>a</i>)
<code>latin1_danish_ci</code>	Danish/Norwegian, case-insensitive
<code>latin1_general_ci</code>	Multilingual, case-insensitive
<code>latin1_general_cs</code>	Multilingual, case-sensitive
<code>latin1_german1_ci</code>	German DIN-1
<code>latin1_german2_ci</code>	German DIN-2
<code>latin1_spanish_ci</code>	Modern Spanish
<code>latin1_swedish_ci</code>	Swedish/Finnish

Table A-2. Comparison of sort order produced by different collations

<code>latin1_swedish_ci</code> (default)	<code>latin1_bin</code>	<code>latin1_general_ci</code>	<code>latin1_german1_ci</code>	<code>latin1_german2_ci</code>
muffin	MX Kollektion	muffin	muffin	Müller
MX Kollektion	MySQL	Müller	Müller	muffin
Müller	Müller	MX Kollektion	MX Kollektion	MX Kollektion
MySQL	muffin	MySQL	MySQL	MySQL

Setting the right sort order

Collation (sort order) can be set at four different levels in MySQL:

- Server
- Database
- Table
- Column

The ability to set collation at these different levels means that it's perfectly acceptable to have a mixture of collations in the same database. So, for instance, if you work in both English and Spanish, you can set the default collation for a database to `latin1_swedish_ci`, but tables or columns that store data in Spanish can be set individually to `latin1_spanish_ci`. As a consequence, all database results will be sorted in the correct order for each language.

Normally, you have no control over changes at the server level unless you are the server administrator. Specifying the sort order for the other three categories—both at time of creation and subsequently—is very easy in phpMyAdmin.

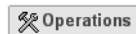
Setting collation at the time of creation

1. Select the required setting from the appropriate Collation drop-down menu at the same time as defining other properties of the database, table, or column.

If you leave the Collation setting blank, the database, table, or column will inherit the collation of its parent. So, a database on a `latin1_swedish_ci` server will automatically use `latin1_swedish_ci` as its collation, but if a table in the same database is set to `latin1_spanish_ci`, all columns within that table will use the Spanish sort order, unless specifically set to a different value.

Changing the collation of an existing database or table

1. Select the database or table in phpMyAdmin, and click the Operations tab at the top of the main frame.



2. When the next screen opens, select the required setting from the Collation drop-down menu, and click the Go button immediately to the right.

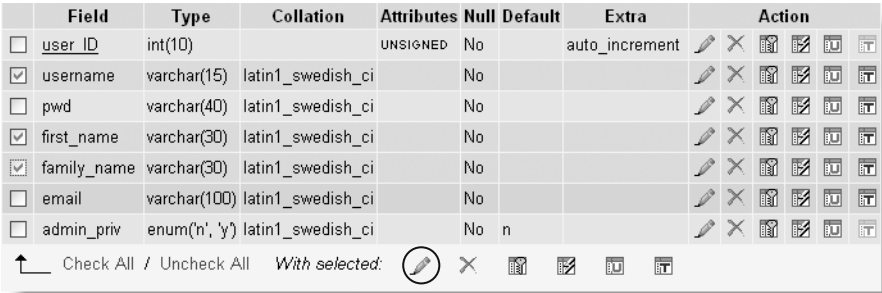


Changing the sort order of a column

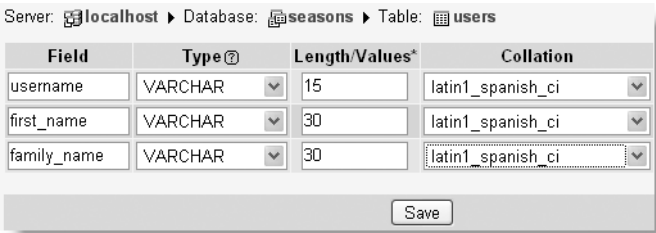
- 1. Use the links in the phpMyAdmin navigation frame to select the table that contains the column or columns you want to change.



- 2. In the table-structure grid in the main frame, place a check mark in the check box to the left of the names of the columns that are to be given a different collation. Then click the pencil icon at the bottom of the grid, as shown here.



- 3. Change the value of the Collation field as required, and click Save.







Appendix B

ESSENTIAL MYSQL MAINTENANCE

One of the joys of working with MySQL is that—in my experience, at least—it requires little or no maintenance. It runs unobtrusively in the background and does its job quietly and efficiently. However, that doesn't mean that you should forget about it entirely. There are some essential tasks that need to be done from time to time. The most important of these is backing up your data. Making a backup is also the way that you transfer your data to another server. This appendix covers the following subjects:

- Backing up your data
- Using a backup file to transfer data to another server
- Updating the privilege tables after upgrading to a later series
- What to do if you forget your root password
- Upgrading and uninstalling MySQL on Mac OS X

Backing up and transferring a database

Something that confuses a lot of people is that you can't just copy a MySQL database file from your hard drive and upload it to your website. Even if you find the right files (on Windows, they're located in `C:\Program Files\MySQL\MySQL Server x.x\data`, where `x.x` represents the MySQL series number), you are likely to damage them unless the MySQL server is turned off. Anyway, most hosting companies won't permit you to upload the raw files, because it would also involve shutting down their server, causing a great deal of inconvenience for everyone.

Nevertheless, moving a database from one server to another is very easy. All it involves is creating a backup “dump” of the data, and loading it into the other database with phpMyAdmin. The dump is a text file that contains all the necessary SQL commands to populate an individual table, or even an entire database, elsewhere. phpMyAdmin can create backups of your entire MySQL server, individual databases, selected tables, or individual tables.

Creating a backup of a single table

- 1. Launch phpMyAdmin and select the database that contains the table you want to back up. Then click the link for the table in the navigation frame.
- 2. When the table details have loaded into the main frame, select Export from the tabs along the top of the screen.



- 3. This opens the rather fearsome looking screen shown in Figure B-1. In spite of all the options, you need concern yourself with only a few.

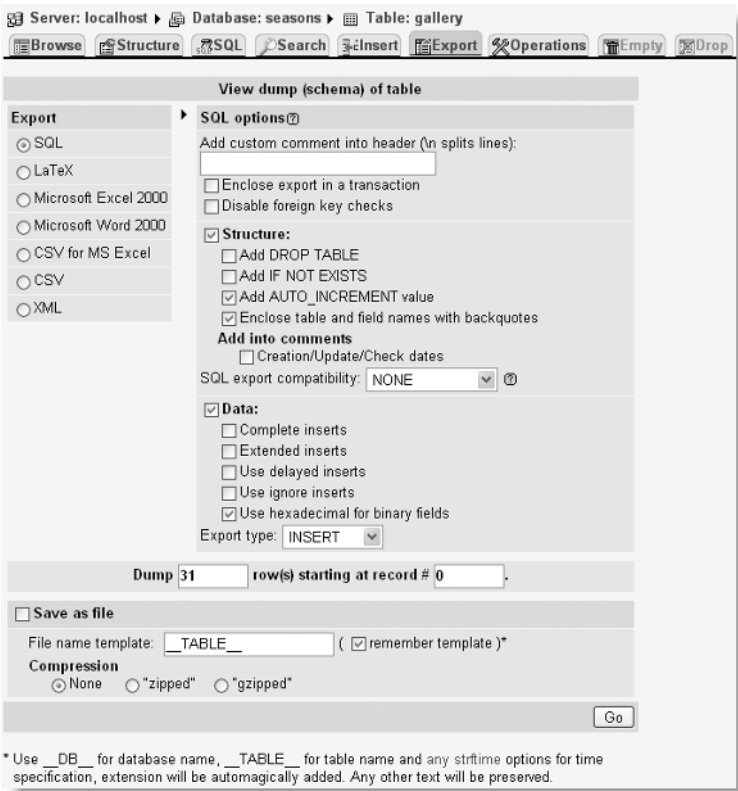


Figure B-1. phpMyAdmin offers a wide range of choice when exporting data from MySQL.

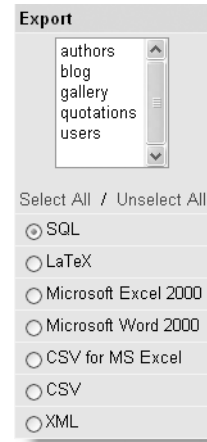
4. The Export section on the left of the screen lists the various export formats available. Unless you're exporting the data to a different program, leave the radio buttons on the default SQL.
5. The only section that needs your attention on the right side of the screen is the middle one labeled Structure. Figure B-1 shows the default settings.
 - If a table of the same name doesn't exist in the database that you want to copy it to, leave the check boxes at their default settings.
 - If the target database already has a table of the same name, select Add DROP TABLE. This will drop the existing table in the target database, and replace it with the data in the backup file.
 - If you have already created the correct table structure on the target database, select Add IF NOT EXISTS. You should use this option only if the target table has exactly the same structure *and* has no existing data.
6. The other important setting in the Structure section is the drop-down menu labeled SQL export compatibility. The setting depends on the version of MySQL running on the other server (only the first two numbers, such as 3.23, 4.0, 4.1 or 5.0, are important):
 - If the other server is running the same version of MySQL, choose NONE.
 - If the other server is running MySQL 3.23, choose MYSQL323.
 - If the other server is running MySQL 4.0, choose MYSQL40.
 - If the transfer is between MySQL 5.0 and MySQL 4.1 (in either direction), choose NONE.
7. Put a check mark in the box alongside Save as file at the bottom of the screen. The default setting in File name template is __TABLE__. This will automatically give the backup file the same name as the table being exported. So, in the case of a table called gallery, it will become gallery.sql. If you add anything after the final double underscore, phpMyAdmin will add this to the name. For instance, you might want to indicate the date of the backup, so you could add 20060228 for a backup made on February 28, 2006. The file would then be named gallery20060228.sql.
8. Click Go to create the backup file. Options for where the file is saved will depend on your browser.

The export processes for multiple tables, a single database, and multiple databases are very similar. However, you should beware of creating very large backup files. Although there is no limit on the size of file that you can export, phpMyAdmin imposes a 2MB limit on loading data from a file.

If you need to transfer a very large database or a collection of databases, you should consult your server's system administrator. Someone with root access (and the requisite knowledge) can move large databases very quickly. If you're interested in learning how to do it yourself, read up on mysqldump at <http://dev.mysql.com/doc/refman/5.0/en/mysqldump.html>.

Backing up a single database or multiple tables from the same database

1. In phpMyAdmin, select the database that you want to export from the drop-down menu in the navigation frame.
2. When the database details have loaded into the main frame, click the Export tab at the top of the screen.
3. The screen that loads is identical to the one shown in Figure B-1, except that the Export section now contains a multiple choice list with the names of all the tables in the selected database, as shown alongside. To export the entire database, click Select All. Otherwise use your mouse pointer in combination with *SHIFT* or *CTRL/⌘* to highlight the tables you require. Leave the radio button on the default SQL.
4. Choose the required options from the Structure section on the right side of the screen. (The meaning of the options is described in steps 5 and 6 of the previous set of instructions.)
5. Put a check mark in the box alongside Save as file at the bottom of the screen. The default setting in File name template is `__DB__`. This will automatically give the backup file the same name as the database being exported. Click Go to create the backup file.



Backing up multiple databases

These instructions are intended principally to help you transfer your local MySQL data to another local computer or to a new location on the same computer. On Mac OS X, it's essential to back up your data and reload it every time you upgrade MySQL. Otherwise, your data will be locked in a redundant folder. Windows users also need to transfer their data when upgrading from one series to another, such as from MySQL 4.1 to MySQL 5.0.

Although you can use these instructions for transferring multiple databases to a remote server, remember the 2MB limit on reloading through phpMyAdmin. Also, deselect the `mysql` database unless you're transferring the data to your own server and understand the consequences of replacing all user accounts on the target server.

1. Launch phpMyAdmin and click the Export link at the bottom of the left-hand column of the phpMyAdmin Welcome page.
2. The screen that opens is similar to the one shown in Figure B-1, but has one extra option, as highlighted in Figure B-2. Use the following settings:
 - **Select All:** Choose this to select all databases. If you are making a backup of MySQL 5.0 or higher, you must use your mouse pointer in combination with *CTRL/⌘* to deselect `information_schema`. The `information_schema` database is new to MySQL 5.0. It's a virtual database that contains information about all the other databases on the server, which MySQL builds automatically. It's a read-only file, so any attempt to overwrite it on the target server will fail.
 - **Add DROP DATABASE:** This will force any database of the same name in the destination location to be dropped and replaced by your backup. If you don't do this, the default `mysql` and `test` databases in MySQL will prevent you from loading data from the backup file. By

copying the mysql database, you can transfer all user accounts and passwords to another server. However, you should do this *before* any new accounts have been created on the server that you are moving to. The DROP DATABASE command completely replaces any existing database with the version from the backup file.

- Set SQL export compatibility: If you are upgrading, the default, NONE, is the correct setting. Otherwise, use the criteria laid out in step 6 of the instructions for backing up a single table.
- Save as file: phpMyAdmin automatically names the backup file after your server. Normally, this will produce a file called localhost.sql.

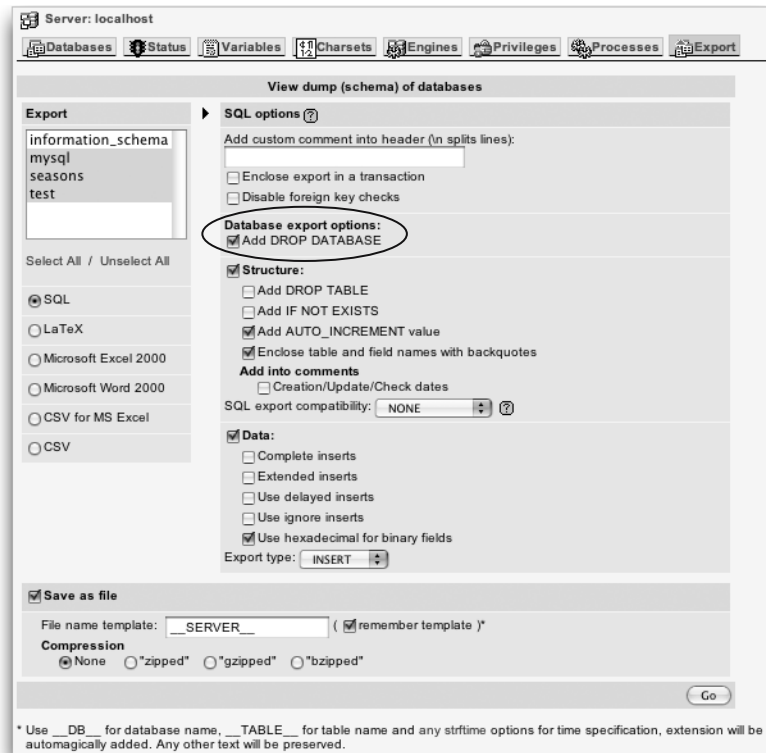


Figure B-2. Selecting the correct options to export all databases

Loading data from a backup file

1. Upload the SQL file to your remote server. (This is not necessary if you are upgrading a local computer or transferring to a new local computer.)
2. If you are transferring a single database, and a database of the same name doesn't already exist on the target server, create the database, but don't create any tables.

3. Launch the version of phpMyAdmin on your target server, and select the database that you plan to transfer the data to. If you are transferring multiple databases, choose any database on the target server. Click the SQL tab in the main frame.
4. Use the Browse button to locate the SQL file, and click Go. That's it!

Or

Location of the text file: (Max: 2,048KB)

Compression:

☒ Autodetect ☐ None ☐ "gzipped"

Character set of the file:

Updating the privilege tables after an upgrade

The instructions in this section are required only if you have upgraded from one series to another and have copied your `mysql` table to the new server.

Updating the privilege tables on Windows

1. Launch the MySQL Command Line Client from the Windows Start button: Programs ► MySQL ► MySQL Server x.x ► MySQL Command Line Client.
2. Enter your MySQL root password when prompted.
3. Type the following command at the `mysql` prompt, and press *ENTER* (replace *x.x* with the number of the MySQL series that you have upgraded to):

```
SOURCE C:/Program Files/MySQL/MySQL Server x.x/scripts/
mysql_fix_privilege_tables.sql
```

Adjust the path in the previous command if you installed MySQL in a different location.

4. Enter your MySQL root password when prompted.
5. You may see some duplicate column warnings as the script runs. These can be safely ignored.
6. When the script has finished running, type `exit` and press *ENTER* to close the MySQL Command Line Client.
7. Stop and restart MySQL for the changes to take effect. (See the “Starting and stopping MySQL manually on Windows” section in Chapter 7.)

Updating the privilege tables on Mac OS X

These instructions assume that you have added MySQL to your environment variables, as described in the “Adding MySQL to your PATH” section of Chapter 7.

1. Launch Terminal from the *Utilities* subfolder of *Applications*.
2. At the Terminal prompt, type the following command followed by *RETURN* (substitute your MySQL root password for *root_password*):

```
mysql_fix_privilege_tables --password=root_password
```
3. You may see some duplicate column warnings as the script runs. These can be safely ignored.
4. When the script has stopped running, stop and restart MySQL using the MySQL pane in System Preferences.

What to do if you forget your root password

Fortunately, it's not as catastrophic as it sounds. You can find the instructions for resetting the root password at http://dev.mysql.com/doc/refman/5.0/en/Resetting_permissions.html. It's relatively straightforward on Windows.

Mac OS X users should open a Terminal window and follow the instructions for Linux, but you will need to invoke superuser privileges through *sudo*. Using *sudo* is simple. You simply precede each command with *sudo* followed by a space. To use *sudo*, you must be logged in as a Mac administrator, and you will be prompted for your Mac password the first time you use *sudo*, or if more than five minutes have elapsed since you last used it. Better still, don't forget your password!

Upgrading and uninstalling MySQL on Mac OS X

Installing MySQL on Mac OS X is incredibly easy, thanks to the PKG files created by MySQL AB, but you can get a very nasty shock (as I did) if you don't plan your upgrades correctly.

According to MySQL, the Mac OS X Installer “does not yet offer the functionality required to properly upgrade previously installed packages.” What this means is that when you “upgrade” MySQL on Mac OS X, you are, in effect, installing a completely new version of MySQL that sits alongside the old version, and, like bad neighbors, they never talk to each other. Apart from the old version taking up unnecessary disk space, that might not be so bad. The problem is that *all your data is locked inside the old version*. Consequently, you should always back up your databases before installing a newer version of MySQL.

Backing up and upgrading on Mac OS X

1. Make a backup of all your databases with phpMyAdmin as described earlier. Make sure that you include the `mysql` database in your backup.
2. Stop MySQL in System Preferences.
3. Install the new version of MySQL. Follow the instructions onscreen.
4. When the installation has completed, restart MySQL.
5. Your new installation of MySQL won't have a root password, so you won't be able to run phpMyAdmin unless you alter `config.inc.php` or create a new root password. A much simpler solution is to load the backup data directly from the command line in a Terminal window.

Copy the backup file to the top level of your home folder.

6. Launch Terminal from the `Utilities` subfolder of `Applications`, and restore your data by issuing the following command followed by *RETURN*:

```
mysql -u root < backup_file
```

Your upgraded version of MySQL should now be working with all previous data in place. If you have upgraded from one MySQL series to another, follow the instructions in the section titled “Updating the privilege tables after an upgrade.”

Removing the older version of MySQL

After installing a new version of MySQL, you may want to remove the old one to save disk space. This section is completely optional, and should not be attempted unless you are comfortable working at the Terminal command line. An old installation of MySQL interferes in no way with the new one.

1. Once you have checked that everything is working correctly with your new installation, you can remove the old files to save disk space. Type the following commands in a Terminal window to change directory and list the files:

```
cd /usr/local  
ls -l
```

2. You should see a list of files as in the following screenshot. There are three references to `mysql`. The first contains `->`, which indicates that it's a symbolic link. The actual name of the folder is `mysql-standard-5.0.13-rc-osx10.4-powerpc` in this example, which is a bit much to expect you to type out every time you want to access `mysql`. So, the symbolic link acts as an alias.

```

Terminal — bash — 80x24
Last login: Tue Oct 18 15:17:07 on ttty1
Welcome to Darwin!
Vigor19:~ davidpowers$ cd /usr/local
Vigor19:/usr/local davidpowers$ ls -l
total 8
drwxr-xr-x  2 root  wheel   68 Jun  3 23:21 lib
lrwxr-xr-x  1 root  wheel   40 Oct 11 17:05 mysql -> mysql-standard-5.0.13-rc-
osx10.4-powerpc
drwxr-xr-x 19 root  wheel  646 Aug 18 12:41 mysql-standard-4.1.14-apple-darwin
8.2.0-powerpc
drwxr-xr-x 19 root  wheel  646 Sep 22 17:26 mysql-standard-5.0.13-rc-osx10.4-p
owerpc
drwxr-xr-x 11 root  staff 374 Apr  4 2005 php5
Vigor19:/usr/local davidpowers$

```

Immediately following are two more references to `mysql`. These contain the actual files. As you can see from the screenshot, the last one has the same number as the symbolic link. It was also created on September 22, so it is more recent than the middle one (which was created on August 18).

To remove the older files, type the following command followed by a space (but don't press *RETURN* yet):

```
sudo rm -r
```

3. Use your cursor to highlight the folder name of the older version, as shown in the following screenshot. Make sure you have the full name, particularly as it's likely to run over two lines. *CTRL*-click and select Paste Selection from the pop-up menu.

```

Terminal — bash — 80x24
Last login: Tue Oct 18 15:17:07 on ttty1
Welcome to Darwin!
Vigor19:~ davidpowers$ cd /usr/local
Vigor19:/usr/local davidpowers$ ls -l
total 8
drwxr-xr-x  2 root  wheel   68 Jun  3 23:21 lib
lrwxr-xr-x  1 root  wheel   40 Oct 11 17:05 mysql -> mysql-standard-5.0.13-rc-
osx10.4-powerpc
drwxr-xr-x 19 root  wheel  646 Aug 18 12:41 mysql-standard-4.1.14-apple-darwin
8.2.0-powerpc
drwxr-xr-x 19 root  wheel  646 Sep 22 17:26 mysql-standard-5.0.13-rc-osx10.4-p
owerpc
drwxr-xr-x 11 root  staff 374 Apr  4 2005 php5
Vigor19:/usr/local davidpowers$ sudo rm -r

```

You should now have a line that looks like this (the actual folder name is likely to be slightly different, depending on when you installed your original version):

```
sudo rm -r mysql-standard-4.1.14-apple-darwin8.2.0-powerpc
```

4. Make sure the command is correct. If it isn't, use *BACKSPACE* to remove anything that shouldn't be there. Equally, if you change your mind, use *BACKSPACE* to remove the entire command.
5. Once you're sure the command is correct, press *RETURN*, and enter your Mac password when prompted. That's it—the old files are gone. And when I say "gone," I really mean it. You cannot restore them from the Trash if you make a mistake.
6. One final bit of housekeeping: You should remove the old version from the Package Receipts directory. Change directory and list the contents by typing the following commands:

```
cd /Library/Receipts  
ls
```

7. The name of the package receipt you need to delete will be the same as in step 3, but with *.pkg* on the end. Remove it by typing this (substitute the correct name for the version on your computer):

```
sudo rm -r mysql-standard-4.1.14-apple-darwin8.2.0-powerpc.pkg
```

Press *RETURN*. If prompted, enter your Mac password.

Hopefully, by the time it comes for you to upgrade, the Mac OS X Installer will have overcome these problems. It's not a lot of work, but it's fiddly and very annoying if you forget to back up your databases first!



INDEX

Symbols

- &: predefined entity, 434
- ': predefined entity, 449
- >: predefined entity, 449, 453
- <: predefined entity, 434, 449
- ": predefined entity, 449
- * (asterisk), as wildcard, 299
- @ (at) character, with XML attributes, 441
- @import statement
 - attaching stylesheets, 111, 116
 - HTML comments with, 116
- ` ` (backticks), in SQL queries, 292
- \ (backslash) character
 - in escape sequences, 145–146
 - with PHP quotes, 62
- , (commas)
 - in arrays, 201
 - in MySQL databases, 256
- \$ (dollar sign), in PHP code, 25, 135
- \$row array element, 266
- \$this variable, cautions, 136
- \$_POST superglobal array, 170–171
- \$_SERVER superglobal array
 - creating intelligent links using, 403–405
 - uses for, 203
 - viewing options for, 25
- \$_SESSION superglobal array, 333, 335
- == (double equal signs), in PHP syntax, 137
- (double minus sign), as decrement operator, 149
- || (double pipes), as logical OR operator, 156
- ++ (double plus sign), as increment operator, 149
- " " (double quotes)
 - in PHP code, 124
 - with text, 143–144
 - with literals, 134
- = (equal sign), as assignment operator, 136–137
- => (equal sign followed by angle bracket) operator
 - creating arrays using, 201
- ! (exclamation point), as negation operator, 156
- != (exclamation point and equal sign), as inequality operator, 154
- !==(exclamation point plus double equal signs), as identical operator, 154
- / (forward slash)
 - in Apache configuration file, 66
 - as division operator, 149
 - in site-root–relative links, 49
- /* */ (forward slash plus asterisk) pair, for PHP comments, 139
- > (greater than) operator, in comparisons, 155
- >= (greater than or equal to) operator, in comparisons, 155
- < (less than) operator, in comparisons, 155
- <= (less than or equal to) operator, in comparisons, 155
- <> (less than, greater than) symbols, as inequality operator, 154
- (minus) button, removing behaviors using, 40
- () (parentheses), with echo and print commands, 143
- % (percent sign), modulo operator, 149
 - creating striped table row displays, 391–392
- . (period) character
 - for linking table and column names (dot notation), 294
 - joining strings (concatenation operator), 146
 - represents current node (XPath), 450
- + (plus) button, 9
 - adding behaviors using, 40
- # (pound or hash sign)
 - for adding comments in PHP, 139
 - in Apache configuration file, 67
- <?php tag
 - function, 134
 - interpretation of, 132
- ;(semicolon), in PHP code, 61, 138
- ' '(single quotes)
 - in PHP code, 124
 - with text, 143–144
- [] (square brackets), in code hints, 132
- === (triple equal signs), as identical operator, 154
- _ (underscore) character, in variable names, 135–136

A

- Access Key option (Input Tag Accessibility Attributes dialog box), 165
- access management
 - folder access restrictions, 374
 - MX Kollektion 3 server behaviors, 365–370, 373–375
 - MX Kollektion User Registration Wizard, 370–373
 - MySQL privileges settings, 245–250
 - page access restrictions, 364
 - password protection approach, 332
 - sessions approach, 332
 - creating PHP sessions, 333
 - destroying sessions, 334
 - implementation, 334–337
 - troubleshooting, 334
 - User Authentication server behaviors, 337, 341–346
 - variables, 333
 - user login system, 362–363
 - user logout system, 365
 - user registration system, 337
 - registration form, 339–346, 349–353
 - storing user data, 338
 - updates and deletions, 353, 356–361
- Access menu (Site Definition dialog box), 91
- accessibility attributes, setting, 106
- action attribute (feedback form)
 - self-processing script, 171
- Action setting (Property inspector), with feedback form, 167
- Add Parameter dialog box (Variables area), 312
- Advanced dialog box (Recordset dialog box), viewing SQL queries, 263
- After start of tag option (Insert drop-down menu), 108
- After tag option (Insert field), 107, 109
- After updating, go to field (Record Update Form Wizard), 279
- aligning, alignment
 - code for, 119, 197
 - dynamically inserted images
 - middle of text placements, 408–411
 - top and bottom placements, 406–408
 - quotations, 267–268
 - Recordset Paging deprecated alignment settings, 277
- all-in-one software packages, 46
- alt attribute, 106
- Alternate text field, 106
- Always show tabs option (Preferences, General category), 20
- anonymous access (MySQL)
 - preventing, 246
 - removing, 235
- Apache web server
 - advantages of using, 52
 - Apache Service Monitor, 56–58
 - background operation, 8
 - Mac OS X systems
 - configuration files, 79
 - preinstallation, 78, 81, 83
 - running IIS in parallel with, 75, 77
 - security advantages, 5
 - software, 4
 - Test Configuration option, 71
 - versions, 52
 - virtual host set up process, 86–88
 - Windows systems
 - configuring to work with PHP, 66, 68–69, 71
 - installation, 53–56
 - starting and stopping program, 56
 - startup preference settings, 58
 - uninstalling, 78
 - upgrading, 52, 77
- Application category (Insert toolbar), 9
 - Record Insertion Form Wizard, 271
- Application Insert bar, 9
 - Record Form Insertion Wizard, 380–381
 - Update Record button, 278
 - XSL Transformation icon, 444
- Application Objects submenu (Insert menu)
 - Show Region server behavior, 188
 - XSL Transformation option, 444
- Apply Comment button (Coding toolbar), 140
- arguments (PHP functions), 134
- arithmetic operators (PHP)
 - combining and assigning, 150
 - precedence, 150
 - summary, 148
 - with SELECT command, 300
- array datatype, 138
- array elements, 266
- arrays
 - array() function, 201
 - array keys, 169
 - array_key_exists() function, 172
 - mixed datatype for, 172–173
 - associative, 170
 - automating navigation menu using, 202–205
 - creating
 - array() function approach, 201
 - associative arrays, 201
 - => operator approach, 201
 - defined, 168
 - for...each loops with, 200–201
 - in navigation menu automation, 202–204
 - indexed, 170
 - multidimensional, 169–170
 - superglobal
 - \$_POST, 170–173
 - \$_SERVER, 25, 203, 403–405

- `$_SESSION`, 333, 335
 - security benefits, 171
 - syntax, 170
 - AS keyword (MySQL), 389
 - assignment operators, 150
 - associative arrays, 170
 - asterisk (*), as wildcard in SQL queries, 299
 - at (@) character, with XML attributes, 441
 - @import statement
 - attaching stylesheets, 111, 116
 - HTML comments with, 116
 - At insertion point option (Insert Div Tag dialog box), 106
 - Attach External Style Sheet dialog box
 - Add to field, 113
 - File/URL field, 113
 - Attach label using for attribute (Input Tag Accessibility Attributes dialog box), 165
 - Attach Style Sheet option (Property inspector, Style menu), 113
 - attachments, sending with email, 185
 - authenticating users, 367–370
 - autocomplete feature, 172
 - autogenerated primary keys, 251
 - autoindent feature, 138
 - automatic code generation feature, 360
 - auto_increment datatype (MySQL), 253
- ## B
- backgrounds
 - adding images to, 114, 116
 - setting using style rules, 392
 - backslashes (\)
 - in escape sequences, 146, 175
 - with PHP quotes, 62
 - unwanted, removing from feedback form, 176–177
 - backticks (` `), in SQL queries, 292
 - Balance Braces tool, 191
 - basename() function, with `$_SERVER` superglobal array, 203
 - bash shell, modifying PATH for MySQL, 232–233
 - basic.css
 - attaching to document, 113
 - code listing, 112–113
 - BBC News (RSS news feeds), 439
 - BBEdit file comparison utility, 27–28
 - Dreamweaver 8 support for, 16
 - problems using, 31
 - Bcc email header (feedback form), 175
 - BETWEEN *min* AND *max* operator, 300
 - Beyond Compare file comparison utility, 27–29
 - Dreamweaver 8 support for, 16
 - problems using, 31
 - binary data (MySQL tables), datatype options, 257
 - Bindings panel
 - accessing, 266
 - dynamic text placeholders, 320
 - expanding Recordset, 266
 - viewing paths for local sources, 447
 - BLOB datatype (MySQL), 257
 - block structure web page layout, 103–104
 - <blockquote> tag, 265–266, 267–269
 - blog dynamic website (blog.php)
 - back-end
 - blog table, 378–379
 - date formatting, 390–391
 - images list, 381–384
 - insert form, 380–381
 - loading blog entries, 387
 - planning and building, 378
 - update and delete forms, 392–396
 - displays
 - accessing full articles, 401–403
 - displaying first section of longer article, 396–397
 - dynamically inserted images, 406–413
 - for empty recordsets, 385–387
 - image storage, 405
 - intelligent links, 403–405
 - main page, 397–400
 - photo galleries, 413, 415–424
 - striped table rows, 391–392
 - visitor comments page, 424
 - bluebells_01.css code listing, 114–115
 - bool (Boolean) datatype, 137
 - box model (Internet Explorer 5/5.5), CSS stylesheets with, 118–119
 - braces, missing, troubleshooting, 189, 191
 - break keyword
 - with switch statements, 194
 - with while and do...while loops, 200
 - browsers
 - previewing styles in, 114
 - viewing web pages in, 109
 - buildImageList() custom function (PHP)
 - creating, 382–384
 - using in update and delete pages, 392–396
 - built-in functions (PHP), 131
 - buttons, inline, adding to web pages, 114, 116
- ## C
- calculations in PHP, 148, 150
 - camel-case notation, 136
 - canceled operations, 281
 - Cannot load mysqli extension error message, 72–73
 - cascading deletes, 288

- cascading stylesheets (CSS), 2
 - attaching, applying
 - accommodating Internet Explorer 5 box model, 118–119
 - adding to styles folder, 112–113
 - @import method, 111, 116
 - to dynamic web pages, 267–268
 - hand-coding versus dialog box approach, 112
 - to feedback form, 167
 - with block layout, 103
 - blog site formatting, 391
 - books about, 112
 - changing automatically, 193, 196–198
 - copying
 - Microsoft conditional comment method, 122
 - PHP includes method, 122–123, 125–126
 - CSS Layout Visualization, 14
 - enhanced support for, 14
 - removing, 117
 - unified CSS panel, 19
 - legacy browser rendering problems, 111
 - photo gallery displays, 416–417
 - removing using trashcan method, 117–118
 - setting backgrounds using, 392
- cascading updates, 288
- case keyword in switch statements, 193–194
- case sensitivity
 - function names, 206
 - XLST, 454
- Cc email header (feedback form), 175
- CGI scripts versus PHP, 167
- change Accessibility preferences link (Insert Div Tag dialog box), 107
- CHAR datatype (MySQL), 255
- characters, legal, in variable names, 135
- Check New Username server behavior (User Authentication), 337, 342–343
- child nodes (XML documents), 440–441
- client-side technology
 - content changes, 6
 - defined, 4
 - PHP request handling process, 7–8
 - validating forms on, 179–184
- client-side XSLT
 - how it works, 434
 - limits, 435
 - tutorial, 435
- closedir() function (PHP), 382
- ⌘+F2 keyboard shortcut (Insert bar), 9
- code listings
 - access management
 - delete user page, 361
 - logging out, 365
 - restricting access to single page, 364
 - update user page, 356–360
 - updating existing user accounts, 356
 - user login page, 362–363
 - user registration form, 341–345, 349–353
- basic.css, 112–113
- bluebells_01.css, 114–115
- blog site, buildImageList() custom function, 383–384
- date() function, format characters, 194–195
- feedback form
 - adding additional email headers, 175
 - \$to, \$subject and \$message variables, 173
 - formatting email input and output, 184–185
 - layout code, 166
 - mail-processing script, 172
 - POST stripslashes snippet, 177
 - preventing email injection attacks, 178
 - style rules, 167
 - thank you messages, 186–190
 - validating on both client and server, 179–184
- insertMenu() function, navigation menu automation, 202–205
- monthly style rules, 196–198
- Code Collapse feature, 15
- code hints
 - enabling and disabling, 25
 - for function arguments, 26
 - square brackets in ([]), 132
 - viewing, 16, 124
 - URL Browser, 124
- Code Inspector (Coding toolbar), 21
- code snippets. *See* snippets
- Code view
 - Coding toolbar, 15, 21
 - Apply Comment button, 140
 - Balance Braces button, 191
 - buttons, 22
 - Collapse Full Tag, 24
 - Expand All button, 24
 - Remove Comment button, 141
 - collapsing and expanding code sections, 15, 22–24
 - enhancements to in Dreamweaver 8, 14–15, 23, 25
 - recordset code, 265
 - Syntax Coloring options, 131
 - Tag Editor, 164
 - XSLT Fragments, 448–450
- coding
 - copying and pasting code, 123
 - and cursor placement, 41
 - fatal versus parse errors, 130
 - third-party file comparison support, 16
- Collapse Full Tag button (Coding toolbar), 24
- Collapse Selection button (Code view), 23

- collation (MySQL)
 - default, 464
 - setting, 465
 - sort order, 464–466
- color coding for dynamically generated output, 95
- columns (MySQL tables)
 - changing sort order, 466
 - datatypes
 - binary data, 257
 - dates and times, 256
 - numbers, 256
 - options, 255
 - predefined lists, 257
 - primary key options, 253
 - text, 255
 - text options, 254–255
 - VARCHAR, 253
 - deleting, 295
 - foreign keys, 290–291
 - naming, 292
- combined assignment operators, 147–148
- Command Line Client (MySQL monitor) passwords, 229
- Commands menu
 - Extension Manager, 103
 - Insert Mark of the Web, 40
- commas (,)
 - in arrays, 201
 - in MySQL databases, 256
- comments
 - in HTML code, conditional, 119, 122
 - in PHP code, 61
 - adding, 139
 - applying and removing, 140–141
- Common Insert bar
 - image icon, 106
 - Insert Div Tag icon, 105
 - Lorem and More extension, 109
- Compare Local Files context menu (Files panel), 30
- comparison operators
 - and case keyword, 194
 - equality, versus assignment operators, 155–156
 - list of, 154–155
 - uses for, 154
 - with SELECT command, 300
 - with WHERE clauses, 300
- CONCAT() function, 308–311
- concatenation operators
 - in PHP code, joining strings using, 146
- conditional comments (Microsoft)
 - copying stylesheets using, 122
 - using with HTML, 111, 119
- conditional regions, 451–453
- conditional statements, 153–154
- Configuration File Path (PHP), recommended setting, 84
- configuring
 - Apache web server, 66, 68–69, 71
 - MySQL
 - collation settings, 465
 - on Mac OS X, security settings, 233–235
 - MySQL Windows Essentials, 218–224, 226–229
 - PHP, 82–85
- Connect button (Apache Service Monitor), 57
- connecting to websites, 259–261
- Connection field (Recordset dialog box), 262
- Connection name field (MySQL Connection dialog box), 259
- Connections folder, including in uploads, 265
- constraints, foreign key, function, 288
- contact forms
 - activating
 - Action setting, 167
 - arrays, 168
 - mail-processing script, 172
 - mail-sending script, 173–174
 - Method setting, 168
 - POST array, 170–171
 - self-processing script, 168, 171, 173, 185
 - automating, third party extensions for, 162
 - creating
 - adding style rules, 167
 - laying out in XHTML, 163–166
 - viewing code, 166
 - modifying
 - adding headers to email messages, 174–175
 - formatting input and output, 184–185
 - preventing email injection attacks, 178
 - removing unwanted backslashes, 176–177
 - validating on both client and server, 179–184
 - thank you message response, 186–190
- content
 - access restrictions, password protection approach, 332
 - content management systems
 - insert forms, 311
 - preventing duplication, 312–315
 - using Update Record server behavior, 315, 319–320
 - static versus dynamic web pages, 6–7
 - updating
 - Delete Record server behavior for, 321–324, 328–329
 - locating incomplete records, 325–326
 - Update Record server behavior for, 315–320
 - using hidden form fields, 326–328
 - using WHERE expression with LEFT JOIN, 325
- Content-Type email header (feedback form), 175
- continue keyword with while and do...while loops, 200
- Control Panel (Windows)
 - adding PHP to Internet Information Services, 73, 75
 - adding PHP to Windows Startup, 65
 - Administrative Tools, Services panel, 227, 229
 - removing MySQL, 224

- control structures (PHP)
 - if...else conditional statements, 153–154
 - switch statements 193–194
 - uses for, 134, 151–153
- copying and pasting code, 123
- copyright notice (copyright.php)
 - adding symbol to web pages, 109
 - control structures, 151–153
 - displaying, 131–133
- copyright protection for news feeds, 439
- Create new database field (phpMyAdmin Welcome screen), 247
- Create New Snippet command (context menu), 192
- CSS. *See* cascading style sheet
- CSS Backgrounds option (Visual Aids menu), 116
- CSS Layout Outlines (Document toolbar, Visual Aids), 106
- CSS Layout Visualization, 14
- CSS panel
 - Attach Style Sheet icon, 113
 - display option, 19
 - enhanced features, 14
 - removing stylesheets from web pages, 117–118
- Ctrl+'/'⌘ '+' keyboard shortcut (balance braces), 191
- Ctrl+Alt+C/Opt+⌘ +C keyboard shortcut (collapse code outside selection), 23
- Ctrl+Alt+E/Opt+⌘ +E keyboard shortcut (expand collapsed code sections), 24
- Ctrl+Alt+I/Opt+⌘ +I keyboard shortcut (insert image), 106
- Ctrl+Alt+J/Opt+⌘ +J keyboard shortcut (collapse code outside full tag), 24
- Ctrl+Alt+T/Opt+⌘ +T keyboard shortcut (insert table), 273
- Ctrl+F10/⌘ +F10 keyboard shortcut (Bindings panel), 266
- Ctrl+F2/⌘ +F2 keyboard shortcut (Insert bar), 9
- Ctrl+F9/⌘ +F9 keyboard shortcut (Server Behaviors panel), 261
- Ctrl+N/⌘ +N keyboard shortcut (new document), 37
- Ctrl+Shift+C/ Shift+⌘ +C keyboard shortcut (collapse selected code), 23
- Ctrl+Shift+J/Shift+⌘ +J keyboard shortcut (collapse full tag), 23
- Ctrl+Shift+R/Shift+⌘ +R keyboard shortcut (Live Data view), 95
- Ctrl+Space keyboard shortcut (code hints), 172
- Ctrl+Z/⌘ +Z keyboard shortcut (undo), 117
- currency formatting, 454
- current year, displaying, 131–133
- current-node dynamic placeholders, 450–451
- cursor location
 - code placement and, 41
 - <div> tag insertions, 106
- custom server behaviors, creating, 346–348

D

- daemons, 8
- data. *See also* databases (MySQL); dynamic websites
 - backing up
 - importance, 469
 - multiple databases, 472–473
 - multiple tables in single database, 472
 - single tables, 470–471
 - complex, breaking into component parts, 251
 - dynamic
 - binding to websites, 266–267
 - viewing, 267
 - icon for, 266
 - displaying, 435
 - loading into tables
 - reloading data from backup files, 473–474
 - SQL file for, 258–259
 - loading onto websites
 - applying the Insert Record server behavior, 310
 - process for, 259–261
 - moving between tables, 292– 93
 - from multiple tables
 - adding content, 304–306, 325
 - combining content from two columns, 308–311
 - content management systems, 311–315
 - creating drop-down menus from, 306–308
 - locating incomplete records, 325–326
 - recordsets using, 297–298
 - server-side XSLT
 - Dreamweaver handling of, 438
 - RSS newsfeeds, 437
 - using a local XML source, 446–448
 - XSLT code structure, 448–460
 - XSLT Fragment creation, 439–443
 - XSLT Fragment embedding, 444–446
 - storing
 - repetitive data, 287
 - in XML documents, 432
 - transferring from MySQL, 212
 - updating
 - using Delete Record server behavior, 321–324, 326–329
 - using hidden form fields, 326–328
 - using Update Record server behavior, 315–320
- Database drop-down menu (phpMyAdmin), 245
- Database field (MySQL Connection dialog box), 260
- database servers, 4
- Database-specific privileges (MySQL), 248
- databases (MySQL)
 - access privileges
 - editing, 250
 - setting in phpMyAdmin, 245–247, 249
 - settings for, 248–250
 - choosing database type, 219–220

- connecting web sites to, 259
 - MySQL Connection dialog box, 259–261
 - troubleshooting, 260
- dates in, 425–428
- features, 213
- fields, multiple, for complex data, 251
- multiple-table
 - adding content, 304–306, 310
 - drop-down menu creation, 306–308
 - combining content, 308–311
 - content management systems, 311–315
 - locating incomplete records, 325–326
 - recordsets, 297–298
 - updating, 315–329
- normalizing process, 296
- primary and foreign keys, 251, 287
- querying, Dreamweaver 8 enhancements, 8, 10
- storing images in, pros and cons, 405
- records and fields, 251
 - expanding results, 276
 - query syntax, 274
 - record navigation system, 276–277
 - reordering results, 274–275
 - viewing, creating table for, 273
- recordset creation, 261–264
- redesigning, 289
- setting up, 247, 249
- storing, 212
- structure of, viewing, 245
- tables
 - column datatypes, 253
 - creating, 252
 - multiple tables, 289–290
 - MyISAM versus InnoDB tablespaces, 219–220
 - datatypes, 253–257
 - populating, 258–259
 - required fields, 253
- updating
 - deleting records, 280–281
 - inserting new records, 270–272
 - updating privileges following, 474–475
 - updating records, 278, 280
- user management
 - user accounts, 248
 - users table, 338
- versus spreadsheets, 252
- Databases panel
 - Server Behaviors panel, 261
 - tree structure, 261
- datatypes
 - mixed, 172–173
 - MySQL databases
 - auto_increment, 253
 - binary data, 257
 - BLOB, 257
 - CHAR, 255
 - DATE, 256
 - dates and times, 256
 - DATETIME, 256
 - DECIMAL, 256
 - ENUM, 257
 - FLOAT, 256
 - INT, 253, 256
 - LONGBLOB, 257
 - MEDIUMBLOB, 257
 - not null, 253
 - numbers, 256
 - predefined lists, 257
 - Primary Key, 253
 - SET, 257
 - TEXT, 255
 - text options, 254–255
 - TIMESTAMP, 256
 - TINYBLOB, 257
 - UNSIGNED, 253
 - VARCHAR, 253, 255
- PHP
 - array, 138
 - Boolean, 137
 - float, 137
 - integer, 137
 - NULL keyword, 138
 - object, 138
 - password, 339, 356
 - resource, 138
 - string, 132, 137
- DATE argument (DATE_FORMAT() function), 389–390
- DATE datatype (MySQL), 256
- date() function (PHP), 143
 - code hint for, 132
 - format characters, 194–195
 - formatting email input using, 184–185
 - versus MySQL DATE_FORMAT() function, 388
 - versus strftime() function, 196
 - with switch statements, 196–198
- dates
 - adding to web pages, 94–95
 - in MySQL tables
 - datatype options, 256
 - formatting and storing, 388–389, 425–428
- DATETIME argument (DATE_FORMAT() function), 389–390
- DATETIME datatype (MySQL), 256
- DATE_FORMAT() function (MySQL)
 - format specifiers, 388–389
 - versus PHP date() function, 388
- DECIMAL datatype (MySQL), 256
- Decision Support (DSS)/OLAP setting (MySQL), 220

- decrement (--) operator, 149
- Default document option (Preferences), 37
- Default Document Type (DTD) option (Preferences), 37
- Default encoding option (Preferences), 37
- Default extension option (Preferences), 37
- Default images folder (Site Definition), 91
- defaults
 - documents, customizing settings for, 37
 - DTD, 32
 - dynamic site file location, 47
 - keyword, with switch statements, 194
 - MySQL
 - communications port setting, 220
 - database type, 219–220
 - resource allocation, 219
 - TEXT datatype, 255
- Define Access Levels dialog box, 364
- Define button (Recordset dialog box), 262
- DELETE command (SQL)
 - basic syntax, 302
 - creating dynamic links for, 320
 - versus DROP command, 295
 - WHERE expression with, 303
- delete form (blog site), 392, 395–396
- DELETE privilege (MySQL), 246
- deleting
 - records from databases
 - cascading deletes, 288
 - Delete Record server behavior, 11, 280–281, 321–324, 328–329
 - snippets, 192
 - users, user accounts, 250, 353, 361
- DESC expression, 301
- Design view
 - and inserting form elements, 164
 - CSS rendering capabilities, 14
 - dotted lines for <div> tags, 106
 - form displays, 164
 - PHP code icon, 132
- Designer layout (Windows Workspace Layouts), 17
- designing and developing websites. *See also* dynamic websites; websites
 - approaches to, 101
 - underlying structure, 103–104
- Developer Machine option (MySQL Server Instance Config), 219
- DHTML (Dynamic HTML), 2
- dial-up connections with remote servers, 96
- directory, server root, pros and cons, 48
- disabling Apache web server, 58
- Disconnect button (Apache Service Monitor), 57
- Display Record Count server behavior, 10
- displays. *See also* dynamic websites
 - blog site
 - accessing full articles, 401–403
 - creating display for empty recordsets, 385–387
 - date formatting, 390–391
 - dynamically inserted images, 406–413
 - extracting first section of longer article, 396–397
 - image storage, 405
 - intelligent link, 403–405
 - loading blog entries, 387
 - main page, 397–400
 - photo galleries, 413, 415–424
 - striped table rows, 391–392
 - forms, 164
 - on remote servers, testing, 45
 - XHTML, MIME type-related problems, 33–34
 - XSLT Fragments
 - parameter-based displays, 457
 - filtered output displays, 455–457
- DISTINCT keyword, 292, 299
- <div> tags
 - blue border, 110
 - dotted line for, in Design view, 106
 - inserting and identifying, 105–106
 - styling web pages using, nested structure, 103–104
 - wrapper, 104
 - in XSLT Fragments, 441
- DIV operator (MySQL), 300–301
- division (/) operator, 149
- do. .while loops, 199
- DOCTYPE declaration
 - changing for existing documents, 38
 - HTML 4.01 Strict, 35
 - HTML 4.01 Transitional, 35
 - importance of, 32
 - new documents, 35–37
 - with PHP code, 39
 - XHTML Mobile 1.0, 36
 - XHTML 1.0 Strict, 35
 - XHTML 1.0 Transitional, 32, 35
 - XHTML 1.1, 35
 - XML declaration with, 40
- Document field (Site Definition), 90
- document structure, importance of, 110
- Document Type (DTD). *See also* DOCTYPE declarations;
DTD definitions
 - changing, 38
 - setting, 105
 - options, 35–37
- document-relative links, 48–49
- documents, tabbed (Mac OS X), 18–19
- dollar sign (\$) in PHP code, 135
- dot notation (.), for linking table and column names, 294

- double numbers, datatype for, 137
 - double quotes (“ ”), in PHP code, 124
 - download sites
 - Apache web server, 53 (Windows)
 - East-West Seasons case study files, 101–102
 - MySQL, 216 (Windows), 230 (Mac OS X)
 - MySQL Administrator, 240
 - MySQL Query Browser, 241
 - PHP, 59 (Windows), 82 (Mac OS X)
 - phpMyAdmin, 237
 - Dreamweaver 8
 - Code view enhancements, 23, 25
 - configuration options, 14
 - CSS support enhancements, 14
 - database connection enhancements, 8
 - DTD-related enhancements, 32
 - coding support enhancements, 15–16
 - file comparisons, support for, 16, 26–31
 - Mac layout options, 18–19
 - new features, 11, 14
 - server behaviors, overview, 9
 - server-side language support, 16
 - Windows layout options, 17
 - XLST support, 16, 432
 - DROP command (SQL), deleting columns using, 295
 - drop-down menus, populating automatically, 306–308
 - DTDs (document type definitions)
 - Dreamweaver 8 options, 32
 - HTML 4.01 Strict, 35
 - HTML 4.01 Transitional, 35
 - importance of, 32
 - setting for new documents, 35–37
 - XHTML 1.0 Strict, 35
 - XHTML 1.0 Transitional, 35
 - XHTML 1.1, 35
 - XHTML Mobile 1.0, 36
 - Dual screen layout (Windows Workspace Layouts), 17, 20
 - dynamic data
 - binding to web sites, 266–267
 - Dynamic Data server behavior, 10
 - icon for, 266
 - Dynamic Extensions (PHP), 63, 85
 - Dynamic HTML (DHTML), 2
 - Dynamic List/Menu dialog box, 307–308
 - Dynamic Page, PHP option (New Document dialog box), 94–95
 - Dynamic Radio Group code (admin_priv field), 356
 - Dynamic Text dialog box, 269
 - dynamic web pages
 - client-side XSLT, 434–435
 - content changes, 5–7
 - how they work, 7–8
 - server-side XSLT
 - advantages, 435
 - code structure, 448–460
 - Dreamweaver handling of, 438
 - RSS news feed display, 437, 439–446
 - using local XML source documents, 446–448
 - dynamic websites, 2. *See also* blog dynamic website, East-West Seasons website
 - binding dynamic data to, 265–267
 - content management systems
 - Delete Record server behavior, 321–324, 328–329
 - insert forms, 311
 - locating incomplete records, 325–326
 - preventing duplication, 312–315
 - Update Record server behavior, 326–328
 - Update Record server behavior for, 315–320
 - using WHERE with LEFT JOIN, 325
 - defined, 3
 - design approaches, 101–104
 - displaying content from multiple columns, 308–311
 - drop-down menus, populating automatically, 306–308
 - formatting
 - adding style rules, 267–268
 - attaching/removing CSS stylesheets, 113–114, 116–119
 - using preset formats, 269
 - Insert Record server behavior with, 310
 - with multiple-table databases, 304–306
 - storing, location for
 - server root approach, 48
 - virtual host approach, 49–50
- ## E
- East-West Seasons website
 - connecting to MySQL database, 259–261
 - main page
 - adding and formatting <div> tags, 106–107
 - adding and linking unordered lists, 108
 - adding copyright symbol, 109
 - attaching/removing CSS stylesheet, 113–114, 116–119
 - creating and naming, 105
 - download files, 101–102
 - replacing filler text and viewing, 109
 - querying MySQL databases, 261–264
 - random quotation generator, 244
 - server-side XSLT displays, RSS news feed example, 437–446
 - echo command, 131
 - displaying output using, 143
 - editing
 - existing user accounts
 - user deletion page, 361
 - user update page, 356–360
 - MySQL user access privileges, 250
 - PHP code, Property inspector for, 133

- php.ini file, 61
- site definitions, 89
- snippets, 192
- SQL queries
 - adding sort criteria, 264
 - limiting number of results, 264
 - Recordset dialog box for, 263
- user accounts, 354–356
- elements
 - in forms
 - inserting, 163
 - in XML source documents, repeating, 441
 - in XSLT code, nested repeated elements, 450–451
 - in XSLT Fragments
 - filtered output displays, 455–457
 - formatting elements, 454–455
 - parameter-based output displays, 457
 - sorting elements, 454
- else clause, wrapping Insert Record server behavior in, 315
- else control structure (PHP), 134, 152–153
- email
 - feedback form
 - Email address check snippet, 180–181
 - formatting input, 184
 - formatting output, 185
 - headers, 174–175
 - injection attacks, preventing, 174, 178
 - sending attachments, 185
- embedding XSLT Fragments in dynamic pages, 444–446
- Enable TCP/IP Networking option (MySQL), 220
- encoding XML documents, 433
- entities, HTML, predefined in XML, 449
- ENUM datatype (MySQL), 257
- Environment Variables (Windows Control Panel), 65
- equal signs
 - double (==), in PHP syntax, 137
 - versus assignment operator, 155–156
 - single (=), as assignment operator, 136–137
 - triple (===), as identical operator, 154
- error messages
 - cannot load mysqli extension, 72
 - configuring PHP to display in Mac OS X, 82–83
 - display options, 129
 - headers already sent, 334
 - MySQL errors, 391
 - PHP
 - external file-related, 131
 - failed to open stream, 130
 - fatal errors, 130
 - notices, 130
 - parse errors, 130
 - warnings, 130
 - when installing Apache web server, 56
- errors, error handling
 - confusing assignment and equality operators, 155–156
 - parse errors, 191
 - preserving input using Sticky Text Field server behavior, 349–352
 - reporting and display options in PHP, 62
- escape sequences
 - in PHP code, 146
 - modifying POST array elements using, 173
- example folders, 102
- exclamation point (!)
 - and double equal signs (!==), as not identical operator, 154
 - and equal sign (!=), as inequality operator, 154
 - as negation operator, 156
- exclusive OR operator (xor), 156
- existing websites, adding dynamic techniques to, 101
- Exit button (Apache Service Monitor), 57
- Expand All button (Coding toolbar), 24
- Export section (phpMyAdmin)
 - backing up multiple databases, 472–473
 - backing up multiple tables in single database, 472
 - backing up single tables, 470–471
- expressions, regular, 178, 396–397
- Extensible Stylesheet Language Transformations. *See* XSLT
- Extension Manager, 103
 - installing snippets collection file, 176
- extensions
 - case study folder, 102
 - Lorem and More, 103
 - snippets collection, installing, 176
 - third-party, for automating contact forms, 162
- external files
 - error messages related to, 131
 - SSIs
 - content changes using, 125–129
 - tips for using, 129
 - testing existence of, logical operators for, 158

F

- failed to open stream error message, 130
- false keyword, 137
 - cautions using, 186
- false values (Boolean datatype), 137
- fatal errors (PHP), 130
- fclose() function (PHP), 382
- feedback forms
 - activating
 - Action setting, 167
 - arrays, 168
 - mail-processing script, 172
 - mail-sending script, 173–174
 - Method setting, 168
 - POST array, 170–171
 - self-processing script, 168, 171, 173, 185

- automating, third party extensions for, 162
- creating, 163–166
- formatting input and output, 184–185
- modifying
 - adding headers to email messages, 174–175
 - inserting POST stripslashes snippet, 177
 - preventing email injection attacks, 178
 - removing unwanted backslashes, 176
- thank you message, 186–190
- validating on both client and server, 179–184
- field, hidden
 - deleting records using, 280–281, 321, 361, 395
 - inserting current date in database, 380
 - updating records using, 316, 318, 328, 356–357
- fields, required, 253
- File Compare feature, 27–31
- File menu
 - New Document dialog box
 - creating PHP documents, 123
 - creating text file, 125
 - Document Type (DTD) options, 35–37
 - Dynamic Page, PHP option, 94–95
 - Preview in Browser option, 96
- FileMerge file comparison utility, 16, 27
- filename extensions
 - .php, 134
 - viewing on Windows, 51
- files
 - comparing
 - File Compare feature, 27–31
 - third-party utilities for, 27
 - customized layouts, 21
 - JavaScript, attaching to web pages, 424
 - locating, 47–48
 - MySQL data file
 - backing up to, 469–473
 - loading data from, 258, 473–474
 - .php filename extension, 134
 - PHP includes, 123–125
 - storage location, virtual host approach, 49–50
- Files panel group
 - Compare Local Files context menu, 30
 - Compare with Remote/Compare with Local context menu options, 30
 - opening Snippets panel, 177
- filler text, creating with Lorem and More extension, 109
- filtering for XSLT Fragment output, 455–457
- firewalls, configuring for MySQL, 223
- Fireworks, 103
- float datatype (PHP), 137
- FLOAT datatype (MySQL), 256
- floating-point numbers, datatype for, 137
- folders
 - for case study, contents, 102
 - parent, 47
 - server root, pros and cons, 48
 - storing images in, 405
 - test folder, 102
- fopen() function (PHP), 382
- for attribute (<label> tag), 165
- for loops, 200
- foreach loops
 - code for, 200–201
 - navigation menu automation, 203
- foreign keys
 - adding to existing tables, 290–291
 - adding to tables, 290
 - and referential integrity, 288
 - constraints, 219
 - purpose, 288
 - using if . . else statements for, 288
 - purpose, 286
 - placing in tables, 290
 - primary keys as, 287
 - transferring from one table to a second table, 294–295
- <form> tags, in feedback form layout, 163–164
- Form fields area (Record Insertion Form dialog box), 271–272
- format specifiers (date() function), 194–195
- Format drop-down menu (XPath Expression Builder), 454
- Format field (Property inspector), 163
- Format menu (Dynamic Text dialog box), preset formats, 269
- FORMAT() function (MySQL), 256
- format-number() function (XPath), using symbols with, 455
- formatting
 - backgrounds, 392
 - blog site display, 391–392
 - dynamic data, 269
 - MySQL dates and time functions, 388–389
 - paragraphs, 109
 - XHTML feedback forms, 163–166
 - XML, XSLT for, 435, 454–455
- forms
 - on blog web site
 - new entry insert form, 380–381
 - update and delete forms, 392–396
- feedback form
 - Action setting, 167
 - activating, 168
 - automating, 162
 - backslashes, removing unwanted, 176–177
 - email injection attacks, preventing, 178
 - formatting input, 184
 - formatting output, 185
 - labeling text fields, 165–166

- laying out, 163–165
- mail-processing script, 172
- mail-sending script, 173–175
- Method setting, 168
- POST array, 170–171
- self-processing script, 171, 173
- style rules, 167
- thank you message, 186–190
- validating on both client and server, 179–184
- insert form
 - adapting for updates, 316–320
 - content management systems with, 311–315
 - for multiple-table databases, 304–306
- user registration form
 - creating, 339–341
 - POST stripslashes snippet, 353
 - Sticky Text Field server behavior, 349–352
 - User Authentication server behaviors, 341–346
- Forms Insert bar
 - Design view and, 164
 - Text Area button, 166
 - Text Field button, 165
- forward slash (/)
 - in Apache configuration files, 66
 - as division operator, 149
 - in site-root–relative links, 49
- forward slash plus asterisk pair (/* */), for PHP comments, 139
- Fragments, XSLT. *See* XSLT Fragments
- friends of ED forum, 46
- FROM keyword (SQL)
 - in recordset creation query, 264
 - with table list, 299
- From email header (feedback form), 175
- F12/Opt+F12 keyboard shortcut (Preview in browser), 96, 109
- fully qualified names (MySQL), 294
- function keyword, 204–206
- function (JavaScript), showImage(), 423
- functions (MySQL)
 - CONCAT(), 309
 - DATE_FORMAT(), 388–389
 - FORMAT(), 256
 - GET_FORMAT(), 389
 - NOW(), 302, 380, 384
 - RAND(), 264, 296–8
 - spaces with, 299, 391
- functions (PHP)
 - code hints for, 26
 - creating, 202–205
 - error messages, 130–131
 - how they work, 382
 - image handling, 411
 - modularizing code using, 205–206
 - passing values to, 206–207
 - returning values to, 208
 - text extraction, 396
 - uses for, 134
 - viewing list of, 172
- functions (PHP built-in)
 - array_key_exists(), 172
 - basename(), 203
 - checkdate(), 427–428
 - closedir(), 382
 - date()
 - code hint for, 132
 - format specifiers, 194–195
 - formatting email input using, 184
 - formatting output, 185
 - versus strftime() function, 196
 - echo, 6, 94–95, 131–132, 143–148
 - empty(), 180, 428
 - error_reporting(), 129
 - fclose(), 382
 - fopen(), 382
 - fread(), 382
 - getdate(), 425
 - getimagesize(), 411
 - get_magic_quotes_gpc(), 177
 - header(), 334
 - htmlentities(), 209
 - include(), 123–126, 129, 152, 157
 - include_once(), 129
 - is_numeric(), 427–428
 - isset(), 182, 190
 - mail(), 173–175
 - mktime(), 257
 - nl2br(), 148, 409
 - opendir(), 382
 - parse_url(), 403–405
 - phpinfo(), 4, 70, 72, 76, 80, 84, 405, 436
 - preg_replace(), 397
 - print, 143
 - readdir(), 382
 - require(), 127–129, 152, 157
 - require_once(), 129
 - session_destroy(), 334, 336
 - session_register() (deprecated), 334
 - session_start(), 333, 335–336
 - session_unregister() (deprecated), 334
 - sha1(), 344, 358, 363
 - strcmp(), 351
 - stripslashes(), 177
 - strip_tags(), 209
 - strlen(), 344
 - strftime(), 196
 - trim(), 180–181

- functions (PHP custom-built)
 - buildImageList(), 382–383, 394
 - getDims(), 411
 - getFirst(), 396–397, 399
 - insertMenu(), 202–204
 - splitText(), 408–410

G

- Generate Image Dimensions server behavior
 - applying, 413
 - creating, 412
- Get image dimensions snippet, 411
- GET method, 168
 - Tag editor default, 305
- getdate() function (PHP), 425
- getDims() custom function (PHP), 411
- getFirst() custom function (PHP), 396–397
 - applying to blog site main page, 399–400
 - image placement using, 408–411
- getimagesize() function (PHP), 411
- GetSQLValueString() (Dreamweaver custom function), 384
- GET_FORMAT() function (MySQL), 389
- Global privileges option (MySQL), 248
- Go to Detail Page server behavior, 10
- gold shield icon, 132–133
- graphical interfaces (MySQL)
 - MySQL Administrator, 240
 - MySQL Query Browser, 241
 - phpMyAdmin, 237–238, 240, 244–245
- greater than (>) operator, in comparisons, 155
- greater than or equal to (>=) operator, in comparisons, 155

H

- hash sign (#)
 - in Apache configuration file, 67
 - in PHP comments, 139
- <head>, copying stylesheets into, 122
- “headers already sent” error message (sessions), 334
- headers, email (feedback form), 174–175
- help, Dreamweaver, 9
- Help menu Extensions Manager, 103
- hidden form fields
 - deleting records using, 280–281, 321, 361, 395
 - inserting current date in database, 380
 - updating records using, 316, 318, 328, 356–357
- hilite class, overriding CSS backgrounds, 392
- hosts, virtual, pros and cons, 49–50
- .htaccess file, 364
 - for directing .html files to .php, 134
- HTML 4 versus XHTML 1.0, 32–33, 35
- HTML 4.01 Strict DTD, 35
- HTML 4.01 Transitional DTD, 35

- HTML comments
 - Microsoft conditional comments, 119
 - with @import statement, 116
- HTML entities in XML, 449
- HTTP address field (Site Definition), 91
- httpd.conf (Apache)
 - altering to work with PHP, 67–68
 - troubleshooting errors in, 70, 72
- hyphens (-), avoiding in table or column names, 292

I

- icons
 - Apache Service Monitor, 56–58
 - for hidden form fields, 280
 - Insert Div Tag, 105
 - lightning bolt, 266
 - for Mac OS X read-only files, 83
 - pencil, 88
 - for PHP code, 132–133
- id attribute (<input> tag), 165
- ID field (Insert Div Tag dialog box), 106
- identical operator (===), uses, 154
- if control structure (PHP), 134
 - adding to copyright.php, 152–153
- if statements (PHP)
 - in form validation code, 183
 - isset() function with, 190
- if . . else statements (PHP)
 - with comparison operators, 154
 - for storing dates in MySQL databases, 426–428
 - as substitutes for foreign key constraints, 288
- IIS (Microsoft Internet Information Services), 49, 52
 - configuring to work with PHP, 73, 75
 - running in parallel with Apache, 75, 77
- image icon (Common Insert bar), 106
- image menu, populating dynamically, 382–384
- Image submenu (Insert menu), 106
- images
 - blog web site images list, 381–384
 - in dynamic web pages
 - getting size dynamically, 411–413
 - inserting and positioning, 106, 406–411
 - photo galleries, 413, 415–424
 - storage decisions, 405
 - PHP functions for handling, 411
- images_blog folder, 102
- images_common folder, 102
- images_gallery folder, 102
- tag
 - alt attribute, 106
 - src attribute, 407–408
- IN() clause with SELECT commands, 301
- Include icon (PHP Insert bar), 123

- include() function
 - versus require, 127–129
- include_once() function, 129
- include_path in PHP error messages, 131
 - when to use, 129
- includes, server-side (PHP)
 - changing content using, 125–129
 - copying and applying stylesheets using, 122–123
 - inserting in web page, 123–124
 - selecting using URL Browser, 124–125
 - testing web pages, 125–126
 - tips for using, 129
- increment (++) operator, 149, 392
- indenting PHP code, 138
- index.php
 - adding to IIS, 75
 - creating in Mac OS X, 80
 - creating PHP include from, 122–123
 - revising to handle data from multiple tables, 297–298
 - saving, 109
- indexed arrays, 169–170
- inequality operator, 154
- information_schema database, 245
- InnoDB tablespace(MySQL)
 - foreign key constraints, 219
 - locating, 220
 - versus MyISAM tablespace, 219–220
- input from users, Sticky Text Field server behavior for, 349–352
- <input> tag
 - in form validation code, 183
 - name and id attributes, 165
 - text field labeling, 165
- Insert bar
 - Application category, 9
 - Application tab, 9
 - display options, 9
 - PHP category, 123
 - XSLT category, 442
- INSERT command (SQL)
 - basic syntax, 301–302
 - when to use, 294
- Insert Div Tag dialog box
 - After start of tag option, 108
 - After tag option, 107, 109
 - Insert field, 106–107
 - New CSS Style button, 106
- Insert Div Tag icon, 105
- Insert field (Insert Div Tag dialog box), 106
- insert forms
 - adapting for updates, 316–320
 - for adding content to multiple tables, 304–306
 - content management systems with, 311–315
- Insert Mark of the Web command (Commands menu), 40
- Insert menu
 - Application Objects submenu
 - XSL Transformation option, 444
 - Show Region server behavior, 188
 - HTML submenu, Special Characters, Copyright option, 109
 - Image submenu, 106
- INSERT privilege (MySQL), 246
- Insert Record server behavior, 11
 - applying to dynamic web sites, 310
 - with user registration form, 340, 352
 - wrapping in else clause, 315
- Insert tab (phpMyAdmin), 254
- Insert field (Insert Div Tag) dialog box
 - After tag option, 107
 - Insert around selection option, 107
- inserting records into databases, 271–272
- insertMenu() function, creating, 202–205
- installing
 - Apache web server on Windows, 52–56, 58
 - MySQL
 - Mac OS X systems, 230–233
 - Windows systems, 214–215
 - MySQL Windows Essentials, 215–218
 - phpMyAdmin MySQL interface, 237–238
 - PHP
 - Mac OS X systems, 78, 80, 83
 - Windows systems, 58–61, 63–64
- INT datatype (MySQL), 253, 256
- integer datatype (PHP), 137
- intelligent links, adding to dynamic web sites, 403–405
- InterAKT, MX Kollection 3 suite, 16, 162, 365–375
- interfaces
 - Macintosh
 - Property inspector display limits, 20
 - tabbed documents, 18–19
 - Windows, preset workspace layouts, 17
 - MySQL, 212
 - graphical, 237
 - MySQL Administrator, 240
 - MySQL monitor, 236
 - MySQL Query Browser, 241
 - phpMyAdmin, 237–238, 240
- Internet & Network (Mac System Preferences), Personal
 - Web Sharing option, 79
- Internet Explorer 5/5.5
 - CSS rendering problems, 111, 118–119
- Internet Explorer 6 SP2, JavaScript blocking, 40
- INTO keyword with INSERT commands, 301
- isset() function
 - in feedback form validation, 182–183
 - with if and elseif statements, 190

J

JavaScript, 2

- attaching external files to web pages, 424
- changing content using, 6
- preventing blocking of, 40
- showImage() function, 423

joint primary keys, 253

K

keyboard shortcuts

- adding images (Ctrl+Alt+I/Opt+⌘+I), 106
- Balance Braces (Ctrl+/,⌘+'), 191
- Bindings panel (Ctrl+F10/⌘+F10), 266
- collapsing and expanding code, 23–24
- CSS panel (Shift+F11—Windows only), 113
- Insert bar display (Ctrl+F2/⌘+F2), 9
- insert table (Ctrl+Alt+T/Opt+⌘+T), 273
- line breaks (Shift+Enter/Shift+Return), 166
- opening new documents (Ctrl+N/⌘+N), 37
- previewing pages in browser (F12/Opt+F12), 96
- restore command (Ctrl+Z/⌘+Z), 117
- Server Behaviors panel (Ctrl+F9/⌘+F9), 261
- Snippets panel (Shift+F9—Windows only), 177
- viewing live data (Ctrl+Shift+R/Shift+⌘+R), 95
- viewing pages in browser (F12/Opt+F12), 109
- viewing PHP functions (Ctrl+Space), 172

L

<label> tag, for attribute, 165

labeling text fields, 165–166

language options (MySQL), 221, 464

last() function (XPath), 451–452

latin1_bin collation (MySQL), 464

latin1_danish_ci collation (MySQL), 464

latin1_general_ci collation (MySQL), 464

latin1_general_cs collation (MySQL), 464

latin1_german1_ci collation (MySQL), 464

latin1_german2_ci collation (MySQL), 464

latin1_spanish_ci collation (MySQL), 464

latin1_swedish_ci collation (MySQL), 464

launch options (MySQL), 221

layout

- customized workspace in Dreamweaver 8, 20–21
- feedback forms, 163–166
- Macintosh tabbed documents, 18–19
- main page, underlying structure, 103–104
- switching between in Dreamweaver 8, 14
- Windows, preset options, 17

LEFT JOIN (SQL)

- locating incomplete records, 325–326
- replacing WHERE expression, 325

legacy browsers, support choices, 111

less than (<) operator, in comparisons, 155

less than or equal to (<=) operator, in comparisons, 155

less than and greater than (<>) symbols, as inequality operator, 154

 tags, in navigation menu automation, 202–203

library (Dreamweaver) items, versus PHP includes, 125

lightning bolt icon, 266

LIMIT command (SQL), 264

with SELECT commands, 301

line breaks, inserting using keyboard shortcuts, 166

<link> tags

adding stylesheets, 113

attaching stylesheets, 111

links

- intelligent, adding to dynamic web sites, 403–405
- to items in PHP pages, 109
- to remote XML files, 440
- site-root-relative versus document-relative, 48–49

lists

- predefined, in MySQL tables, 257
- unordered, adding to pages, 108

literals (PHP), uses for, 134

Live Data view

- enhanced features, 14
- limits, 403
- resource use, 133
- viewing dynamic content using, 267
- with remote testing servers, 95

Liyanage, Marc (Mac OS X PHP upgrade utility), 81–82

local files, comparing, 28, 30

Local Info category (Site Definition)

- Default images folder field, 91
- Document field, 90
- HTTP address field, 91
- Site name option, 90

local testing servers

- checking for XSL extension support, 436
- connecting to MySQL databases using, 259
- pros and cons, 44–45
- setup
 - Mac OS X, 78–83
 - software needs, 45–46
 - virtual hosts and directories, 49–50, 86–89
 - Windows systems, 51–52, 55, 58, 60–61, 63–64, 66, 68–73, 75–77
- site definition, 89–94
- site testing, 94–95

local XML source documents

- embedding table displays in, 448
- setting up, 446–447

Local/Network option (Site definition), 91

localhost option (MySQL Connection dialog box), 259

Locate XML Source for XSL Template dialog box, 447

- Log In User server behavior
 - Dreamweaver, 337, 362–363
 - MX Kollection 3, 367–370, 372–373
- Log Out User server behavior
 - Dreamweaver, 337, 365
 - MX Kollection 3, 373
- logical operators
 - OR operator, 156
 - with SELECT command, 300
 - uses for, 156–158
- Login Information table (phpMyAdmin), 248
- login page
 - cautions applying Restricted access to, 364
 - MX Kollection Login Form Wizard for, 372–373
 - User Authentication server behaviors for, 362–363
- Long description field, 107
- LONGBLOB datatype (MySQL), 257
- longdesc attribute settings, 107
- Looper Wizard (MX Kollection), applying to photo gallery display, 420–421
- loops
 - for loops, code for, 200
 - foreach loops, code for, 200–201
 - in navigation menu automation, 202–204
 - while and do...while, code for, 199
 - in XSLT code, 450
- Lorem and More extension, 103
 - creating dummy blog entries, 387
 - replacing filler text using, 109

M

- Mac OS X systems
 - default Apache installation, 81
 - Dreamweaver 8 on, 14
 - file comparison utilities, 27
- MySQL
 - adding to PATH variable, 232–233
 - case sensitivity, 236
 - downloading and installing, 230–231
 - privilege tables, updating, 475
 - security settings, 233–235
 - upgrading and uninstalling, 475–478
- PHP setup
 - test server setup, 78, 80, 82–83
 - upgrading, 80–82
 - viewing configuration files, 84–85
- Property inspector display problems, 20
- server root folder, 47
- sharing files with Windows systems, 21
- tabbed document feature, 18–19
- virtual hosts, registering, 88
- magic quotes
 - disabling, 176, 353
 - in Dreamweaver, 177
 - in PHP code, 62
- mail() function
 - feedback form
 - adding headers to email messages, 174–175
 - testing, 173
 - troubleshooting, 174
 - passing POST array variables to, 173
- mail-processing script (feedback form)
 - adding to action attribute, 172
 - superglobal arrays, 170–171
- mail-sending script (feedback form)
 - adding headers, 175
 - creating, 173
 - testing, 173
 - troubleshooting, 174
- main page
 - blog.php site, 397–400
 - returning to using intelligent links, 403–405
 - using URL parameters in, 401–403
 - East-West Seasons site
 - adding and formatting <div> tags, 106–107
 - adding and formatting unordered lists, 108
 - adding copyright symbol, 109
 - attaching/removing CSS stylesheet, 113–114, 116–119
 - creating and naming, 105
 - replacing filler text and viewing, 109
 - layout, underlying structure, 103
- main() references, error messages with, 131
- maincontent <div>
 - filler text, 107
 - formatting options, 107
 - locating nav <div> in relation to, 108
 - replacing text in, 163
- Master Detail Page Set server behavior, 11
- MEDIUMBLOB datatype (MySQL), 257
- menus
 - drop-down, populating automatically, 306–308
 - image list, 382–383
 - navigation menu, automating using arrays, 202–205
- Method setting
 - Property inspector, using with feedback form, 168
 - Tag editor, 305
- MIME information, adding to email output, 185
- MIME types, and XHTML 1.0 display problems, 33–34
- minus (-) button, removing behaviors using, 40
- minus sign, double (--), as decrement operator, 149
- mixed datatypes, 172–173
- Module Settings (PHP), 64

- modulo division (%) operator, 149
 - creating striped table row displays, 391–392
- monitor resolution, Macintosh, Property inspector display limitations, 20
- monitor, MySQL, using, 229, 236
- Move to New Window (Macintosh), 19
- multidimensional arrays, 169–170
- Multifunctional Database (MySQL configuration on Windows), 219
- multiple conditional regions, 452–453
- multiple tables
 - splitting off existing tables, 289–291
 - transferring data, 291–293
- MX Kollektion 3 (InterAKT)
 - Login Form Wizard, 372–373
 - MX Looper server behavior, 415–417, 420–421
 - MX Send Email extension, 162
 - Recordset dialog box, QuB3 button, 385
 - splitting off existing tables, 289–291
 - user management server behaviors, 365
 - installing, 366–367
 - Logout User server behavior, 373
 - pros and cons, 374–375
 - Restrict Access to Folder, 374
 - setting up for user authentication, 367–370
 - User Registration Wizard, 370–372
- MX Send Email (InterAKT), 162
- MyISAM tables (MySQL)
 - foreign key constraints not supported, 288
 - selecting as default table type, 226–227
 - versus InnoDB tablespace, 219–220
- MySQL
 - advantages of using, 212
 - background operation, 8
 - collation
 - setting, 465
 - sort order, 464–465
 - components, 213–214
 - costs, 4
 - cross-platform capability, 5
 - data backups, 469
 - multiple databases, 472–473
 - multiple tables in single database, 472
 - reloading data from backup files, 473–474
 - single tables, 470–471
 - databases
 - accessing from websites, 261
 - connecting websites to, 259–261
 - datatypes, 253–259
 - date storage, 425–428
 - foreign key constraints, 288
 - populating tables, 258–259
 - privileges settings, 245
 - records and fields, 251
 - recordset creation, 261–264
 - setting up, 247, 249
 - tables, 252–253
 - viewing list of, 245
 - viewing structure of, 245
 - functions. *See also* functions (MySQL)
 - troubleshooting errors with, 391
 - white spaces and, 299
 - InnoDB tablespace, locating, 220
 - interfaces
 - graphical, 237
 - MySQL Administrator, 240
 - MySQL monitor, 236
 - MySQL Query Browser, 241
 - phpMyAdmin, 237–238, 240, 244
 - on Mac OS X systems,
 - installing 230, 232–233
 - security settings, 233–235
 - upgrading, 475–478
 - names, fully qualified, 294
 - non-English languages and character sets, 463
 - online documentation, 461
 - as open source technology, 5
 - phpMyAdmin settings, 245
 - queries, 288
 - rationale for choosing, 4
 - reserved words, viewing list of, 292
 - security
 - advantages, 5
 - shared hosting, 246
 - tables
 - deleting columns from, 295
 - multiple-table databases, 304–311
 - revising, 290–291
 - TIMESTAMP column, 379, 388
 - updating privileges table following upgrade, 474–475
 - user accounts
 - creating, 246
 - managing, 245
 - privilege settings, 246–250
 - setting up in phpMyAdmin, 248
 - versions
 - backwards compatibility, 213
 - 5.0, 215, 245
 - and Mac OS X installations, 230–231
 - removing 4.1.5 and earlier, 214–215
 - Windows Essentials, 214–229
 - on Windows systems
 - installing, 214–219
 - removing and upgrading, 224
 - running, 229
 - security settings, 222
- MySQL Command Line Client file (Windows), 474
- MySQL connection collation setting (phpMyAdmin), 245

- MySQL Connection dialog box, 259
 - Connection name field, 259
 - Database field, 260
 - MySQL server field, 259
 - Password field, 260
 - Username field, 260
- mysql database, 214
 - function, 245
 - viewing structure, 245
- MySQL Query Browser interface, 240–241
- MySQL server field (Connection dialog box), 259
- MySQL Server Instance Config Wizard (Windows), 218–224
- MySQL Windows Essentials
 - configuring, 218–219
 - changing default table type, 226–227
 - changing to manual startup, 227, 229
 - database type options, 219–220
 - Decision Support (DSS)/OLAP setting, 220
 - Enable TCP/IP Networking option, 220
 - InnoDB tablespace location, 220
 - language settings, 221
 - launch options, 221
 - MyISAM versus InnoDB tables, 219
 - reconfiguring, 224
 - security settings, 222–223, 233–235
 - installing, 215–218
 - running two series in parallel, 225–226
- mysqld daemon, 214

N

- name attribute (<input> tag), 165
- Name field (Recordset dialog box), 262
- Name option (Input Tag Accessibility Attributes dialog box), 165
- naming
 - fully qualified names (MySQL columns), 294
 - functions, 205–206
 - PHP variables, 135–136
 - recordsets, 262
 - tables or columns, 292
- navigation
 - automating
 - code for, 202–205
 - loops for, 199
 - for database records, 276–277
 - for MySQL, 244–245
 - for photo gallery, adding to display, 421
- negation operator (!), 156
- negation testing, logical operators for, 156–157
- nested repeated elements (XML nodes), 450–452
- NetInfo Manager (Mac OS X), registering virtual hosts, 88
- Netscape 4, CSS-rendering problems, 111

- New Document dialog box (File, New menu)
 - Dynamic Page, PHP option, 94–95
 - for text files, 125
 - for XSLT Fragments, 439
- New Site option (Site menu), defining a PHP site, 89–90
- news feeds
 - adding to web pages, 437
 - how it works, 438
 - RSS feeds, 437
 - choosing, factors to consider, 439
- newuser.php (user registration form)
 - creating, 339–341
 - Sticky Text Field server behavior, 349–352
 - User Authentication server behaviors, 341–346
 - user deletions, 361
 - user updates, 356–360
- nl2br() function, 148, 403
- nodes (XML source documents), 440–441
- Non-Transactional Database Only (MySQL configuration on Windows), 219
- nonbreaking space entity (), 125
- None option (Document Type drop-down menu), 35
- normalization process (databases), 296
- Norton Internet Security (NIS)
 - Apache web server installation, 55
 - MySQL with, 223
- not identical operator (!=), 154
- not null setting (MySQL), 253
- notice messages (PHP), 130
- NOW() function (MySQL), 380
- NULL keyword (PHP), 137–138
 - how Dreamweaver treats NULL in MySQL, 253
- numbers. *See also* operators
 - datatypes for, 137 (PHP), 256 (MySQL)
 - in PHP variables, 137

O

- object datatype, 138
- OOP (object-oriented programming) features in PHP, 136
- open source technology, 5
- opendir() function (PHP), 382
- Opera 8.5, and client-side XSLT, 435
- Operations tab (MySQL), changing MySQL collation, 465
- operators
 - arithmetic, 148
 - combining and assigning, 150
 - precedence, 150
 - assignment, 136–137
 - combined assignment, 147–148
 - comparison, 154–155
 - concatenation, 146
 - conditional, 154–156
 - logical, 156–158

- modulo, 149, 391–392
- with SELECT command, 300
- ternary, 426–428
- with WHERE clauses, 300
- optimizing databases, normalization process, 296
- optional fields, MySQL tables, 253
- or keyword, as logical OR operator, 156
- ORDER BY command (SQL)
 - RAND() function with, 264
 - in recordset creation, 264
 - with SELECT commands, 301
 - sorting recordsets using, 263–264
- output
 - date() versus strftime() functions, 196
 - PHP text handling, 146–148
 - XSLT Fragments
 - filtered displays, 455–457
 - sending parameters from PHP pages, 458–460
 - using XSLT parameters, 457

P

- <p> tags, server-side includes with, 125
- paging recordsets, 276–277
- panel groups, altered Mac display for, 18–19
- paragraphs
 - adding to feedback forms, 164–165
 - formatting text as, 109
- parameters. *See also* functions
 - URL, accessing data using, 401–403
 - XSLT
 - displaying output using, 457
 - sending from PHP pages, 458–460
- parent folders, 47
- parent nodes (XML source documents), 440
- parentheses (), with echo and print commands, 143
- parse errors
 - missing braces, 189, 191
 - versus fatal errors, 130
- parse_url() function (PHP), for intelligent links, 403–405
- parsing
 - and site location, 47
 - XML, Unicode encoding, 433
- Password field (MySQL), 248, 260
- passwords
 - encrypting with sha1(), 344, 358, 363
 - Mac OS X, MySQL root password, 234–235
 - MySQL
 - setting using phpMyAdmin, 248
 - for MySQL monitor, 229
 - password data type
 - cautions, 356
 - user registration form, 339

- reissuing forgotten password, 363, 373
- root, resetting, 475
- patch level (pl), 237
- PATH variable (Mac OS X)
 - adding MySQL to, 232–233
 - modifying in bash shell, 232–233
 - modifying in tcsh shell, 233
- PDO (PHP Data Objects), 16
- pencil icon, 88
- percent sign (%)
 - creating striped table row displays, 391–392
 - as modulo division operator, 149
- performance, in Live Data view, 133
- period (.) character
 - as concatenation operator, 146–147, 173
 - in code for current nodes, 450
 - in qualified column names (MySQL), 294
- photo gallery, displaying on blog site, 413, 415
 - activating thumbnails, 423–424
 - main image area, 418–422
 - populating gallery table, 417
 - tables and style rules, 416–417
- PHP
 - arrays
 - associative, 170
 - indexed, 170
 - multidimensional, 169–170
 - superglobal, 170–171
 - syntax for, 168
 - auto completion feature, 25
 - calculations, 148, 150
 - code
 - code hints, 25
 - comments, 139–141
 - cursor placement cautions, 41
 - gold shield icon for, 132–133
 - indenting, and white spaces, 138
 - syntax, 137–138
 - variables, 134–137
 - combining with XHMTL, 182–184
 - comparison operators
 - equality versus assignment operators, 155–156
 - list of, 154–155
 - uses for, 154
 - configuring, 84–85, 353
 - control structures, 134
 - if . . else conditional statements, 153–154
 - switch statements, 193–194
 - uses for, 151–153
 - costs, 4
 - cross-platform compatibility, 5

- datatypes
 - array, 138
 - Boolean, 137
 - float, 137
 - integer, 137
 - NULL, 138
 - object, 138
 - resource, 138
 - string, 137
- DOCTYPE declaration, impact of code above, 39
- Dynamic Extensions (Windows), 63
- enhanced support for in Dreamweaver 8, 16
- error messages
 - external file-related, 131
 - fatal errors, 130
 - notices, 130
 - parse errors, 130
 - warnings, 130
- functions. *See* functions (PHP), functions (PHP built-in)
- if . . else statements with, as substitute for foreign key constraints, 288
- includes, 122–123
- literals, 134
- logical operators, 156–158
- on Mac OS X systems
 - enabling and configuring, 80
 - error display, 82–83
 - installation, 78, 83
 - upgrading, 80–82
- magic_quotes_gpc, 176
- Module Settings, 64
- OOP features, 136
- as open source technology, 5
- .php filename extension, 134
- <?php tag, 132, 134
- rationale for choosing, 4
- request processing, 7–8
- resetting error reporting and display options, 62
- security advantages, 5
- sending XLST parameters from, 458–460
- sessions
 - creating, 333
 - destroying, 334
 - implementation, 334–337
 - troubleshooting, 334
 - variables, 333
- single versus double quotes, 124
- site definition in Dreamweaver 8, 89–95
- strcmp() function, 352
- templates with, 6
- text handling
 - adding to strings, 147–148
 - escape sequences, 146
 - joining strings, 146
- version 5.1, 16
 - on Windows systems
 - adding to startup procedure, 65
 - configuring Apache for, 66, 68–69, 71
 - configuring IIS for, 73, 75
 - installing, 58–61, 63–64
 - uninstalling, 78
 - upgrading, 59, 77, 80–82
 - XSLT support, 436
- PHP Core settings, recommended values, 84
- PHP documents, creating, 123
- PHP includes
 - changing content using, 125–126
 - copying and applying stylesheets, 125–126
 - Dreamweaver limitations, 204
 - include() versus require(), 127–129
 - tips for using, 129
 - versus templates or library items, 125
- PHP Insert bar, Require button, 127
- PHP MySQL option (Site Definition), 91
- phpinfo() function
 - testing support for XSL Extensions using, 436
 - viewing PHP configuration information, 84–85
- php.ini file
 - comments, 61
 - creating, 60
 - editing, 61
 - error notices, disabling, 351
 - modifying during Windows setup, 61–64
 - register_globals, importance of, 170
 - saving sessions in, 337
 - SMTP settings (Windows only), 173
 - viewing, 60, 84–85
- PHPMailer utility, 185
- phpMyAdmin (MySQL interface)
 - accessing, 244
 - backing up MySQL data, 469
 - multiple databases, 472–473
 - multiple tables in single database, 472
 - single tables, 470–471
 - changing MySQL collation, 465
 - column sort order, changing, 466
 - creating tables, 250–257
 - Database drop-down menu, 245
 - database setup, 247, 249
 - database structure, viewing, 245
 - downloading and installing, 237–238
 - loading data from backup files, 258, 473–474
 - main frame, 244
 - navigation frame, 244
 - SQL tab, viewing and running SQL queries from, 291
- tables
 - blog table, 378–379
 - creating, 252–253
 - datatypes, 253–257

- defining foreign key column, 290
 - deleting columns, 295
 - entering records, 254
 - foreign keys, 290–291
 - multiple tables, 289–290
 - populating using SQL files, 258–259
 - text datatypes, 255
 - users table, 338
 - troubleshooting, 240
 - user accounts
 - creating, 246
 - managing, 245
 - privilege settings, 246–250
 - Welcome screen
 - Language setting, 245
 - Privileges setting, 245
 - pipes, double (| |), as logical OR operator, 156
 - placeholders
 - for dynamic content, 266–267, 451
 - for dynamically inserted images, 407
 - for photo gallery images, 420
 - plus (+) button (Server Behaviors panel), 9
 - adding behaviors using, 40
 - plus sign, double (++), as increment operator, 149
 - Position option (Input Tag Accessibility Attributes dialog box), 165
 - position() function, in XSLT conditional region code, 451–452
 - POST array (\$_POST), 168
 - checking for array keys, 172
 - in feedback form, 170–171
 - modifying elements in, 173
 - with Tag editor, 305
 - POST stripslashes snippet
 - inserting into feedback form, 177
 - with user registration form, 353
 - pound sign (#), in Apache configuration file, 67
 - precedence
 - arithmetic operators, 150
 - logical operators, 156
 - precompiled software packages, pros and cons, 46
 - predefined lists (MySQL tables), datatypes for, 257
 - predefined variables (superglobal arrays), 170–171
 - Preferences, Dreamweaver
 - accessibility features, disabling, 106–107, 165
 - accessing code hints, 25
 - Always show tabs option, 20
 - default document setting options, 37
 - layout options, 20
 - registering third-party file comparison utilities, 27
 - preg_replace() function (PHP), 397
 - preset formats and layouts
 - applying to dynamic web pages, 269
 - Windows systems, 17
 - Preview in Browser option (File menu), 96
 - primary keys
 - identifying in tables, 251–252
 - locating foreign keys in relation to, 290
 - Primary Key datatype (MySQL), 253
 - using as foreign key in related tables, 287
 - print command versus echo command, 143
 - Privileges (phpMyAdmin), 248
 - privileges table (MySQL), updating after upgrades, 474–475. *See also* access management
 - Problems and Common Errors page (MySQL online documentation), 461
 - .profile file (Mac OS X), adding MySQL to PATH, 232–233
 - Property inspector
 - Action setting, 167
 - Format field, 109, 163
 - Mac OS X systems, 20
 - Method setting, 168
 - Style drop-down menu, 113
 - Unordered List button, 108
 - viewing and editing PHP code, 133
 - punctuation, importance of in SQL queries, 299
- ## Q
- QuB3 button (MXKollection Recordset dialog box), 385
 - queries (SQL)
 - case insensitivity, 299
 - DELETE command
 - basic syntax, 302
 - WHERE expression with, 303
 - creating list of records, 274–276
 - creating using Recordset dialog box, 10
 - Dreamweaver 8 enhancements, 8
 - editing, 264
 - INSERT command, basic syntax, 301–302
 - moving data between tables, 291–293
 - in MySQL monitor, 236
 - ordering SQL commands in, 299
 - primary and foreign keys, 288
 - punctuation, 299
 - for recordset creation, 261–264
 - SELECT command
 - arithmetic operators with, 300
 - basic syntax, 299
 - BETWEEN min AND max with, 300
 - DIV clause with, 301
 - FROM command with, 299
 - IN() operator with, 300–301
 - LIMIT clause with, 301
 - logical operators with, 300
 - with multiple tables, 297–298
 - ORDER BY clause with, 301
 - syntax, 299
 - WHERE clause with, 299–300

- sorting results, 263
 - storing recordsets, 264
 - transferring foreign keys, 294–295
 - UPDATE command
 - basic syntax, 302
 - WHERE expression with, 302
- <?php tag
 - function, 134
 - interpretation of, 132
- quotes
 - single versus double in PHP code, 124, 143–144
 - using with literals, 134
- R**
- RAND() expression in ORDER BY clauses, 264, 301
- random order generation
 - adding to SQL queries, 264
 - quotation generator, 244
- readdir() function (PHP), 382
- Record column (Test SQL Statement panel), sorting
 - records using, 262
- Record Insertion Form Wizard
 - accessing, 271, 81
 - dialog box options, 271–272
- Record Update Form Wizard
 - accessing, 278
 - updating records, 279–280
- records (MySQL), 251
 - constraints, modifying, 288
 - incomplete, locating, 325–326
 - deleting, 280–281
 - inserting, 254, 270–272
 - in multiple-table databases, adding data, 304–306
 - updating, 278, 280
 - viewing list of
 - creating table for, 273
 - expanding results, 276
 - navigation system for, 276–277
 - query syntax, 274
 - reordering results, 274–275
- Recordset dialog box
 - Advanced dialog box
 - viewing SQL queries in, 263
 - working with multiple tables, 297–298
 - Server Behaviors panel
 - Connection field, 262
 - DEFINE button, 262
 - Name field, 262
 - Recordset, 10
 - Recordset Paging, 10, 276–277
 - Table drop-down menu, 262
 - updating records, 278, 280
- Recordset dialog box (MX Kollection), QuB3 button, 385

- Recordset Navigation Bar, 421
- recordsets
 - creating, 261–264
 - naming, 262
 - sorting, 262
 - SQL queries for, 263
 - defined, 10
 - for dynamic web sites, automatic menu-population, 306–308
 - limiting number of items in, 264
 - revising, 297–298
 - sorting, 263–264
 - storing, 264
 - viewing in Bindings panel, 266
- redesigning relational databases, 289
- redirecting .html files to .php, 134
- referential integrity, 288
- register_globals setting (php.ini), 170
- regular expressions, 178, 396–397
- relational databases
 - features, 250
 - fields, 251
 - multiple tables, foreign keys, 286
 - redesigning, 289
- Relative to field (URL Browser), 124–125
- remote files
 - comparing, 30
 - XML, linking to, 440
- Remote Info category (Site Definition), 91, 93–94
- remote servers, 4
 - connecting to MySQL databases using, 259
 - testing servers
 - display capabilities, 45
 - using, 95–96
 - Windows systems, 224
 - transferring multiple databases to, 472–473
 - upgrading MySQL files from, 473–474
- Remove Comment button (Coding toolbar), 141
- removing
 - MySQL
 - MySQL versions during set up, 214–215
 - from Windows systems, 224
 - stylesheets, 117
- Repeat Region (XSLT), 442
- Repeated Region server behavior, 10, 275–276
- repeating elements
 - in XML source documents, 441
 - in XLST fragments, 442
- repetitive tasks, custom server behaviors for, 346
- Require button (PHP Insert bar), 127
- require() function versus include() function, 127–129
- required fields (MySQL tables), 253
- require_once(), 129
- reserved words (MySQL), viewing list of, 292

- resource datatype, 138
- Restore section (MySQL Administrator), 240
- Restrict Access to Folder server behavior (MX Kollection 3), 374
- Restrict Access to Page server behavior (User Authentication), 337, 364
- rollover effect, creating using CSS, 104
- root administrator (MySQL), 222, 233
- root element (XML source documents), 440
- root password (MySQL)
 - resetting, 475
 - setting on Mac OS X systems, 234–235
 - setting on Windows, 222
- root user (MySQL), cautions, 246
- RSS news feeds
 - displaying using server-side XSLT, 437–438
 - structure, 440

S

- saving customized workspace layouts, 20–21
- script editors, 61
- scripts
 - fatal versus parse errors, 130
 - mail-processing, 172
 - mail-sending
 - code for, 173
 - testing, 173
 - troubleshooting, 174–175
 - self-processing
 - adding to action attribute, 171
 - testing, 173
- security
 - access management
 - folder access restrictions, 374
 - logging in users, 362–363
 - logging out users, 365
 - MX Kollection 3 server behaviors, 365–370, 373–375
 - MX Kollection User Registration Wizard, 370–373
 - page access restrictions, 364
 - password protection approach, 332
 - sessions approach, 332–337
 - User Authentication server behaviors, 337, 341–346
 - user registration system, 337–346, 349–353
 - user updates and deletions, 353, 356–361
 - and Apache web server installation, 55
 - MySQL
 - Mac OS X settings, 233–235
 - shared hosting, 246
 - Windows settings, 222–223
 - passwords, root, resetting, 475
 - register_globals setting, 170–171
 - sessions, 332
 - validation tools

- preventing email injection attacks, 178
- validating forms on client and server, 179–184
- SELECT command (SQL)
 - basic syntax, 299–300
 - CONCAT() function with, 308–311
 - content management using, 312–315
 - with LEFT JOIN, 325
 - moving data between tables using, 291–293
 - querying multiple tables, 297–298
 - in recordset creation query, 264
- select list (SELECT commands), 299
- SELECT privilege (MySQL), 246
- Select record from field (Record Update Form Wizard), 279
- self-processing forms
 - contact form, 167–168
 - feedback form
 - adding to action attribute, 171
 - testing, 173
 - variables, duration of, 185
- semicolon (;), in PHP code, 61, 138
- server behaviors
 - adding and removing, 40
 - custom, creating, 346
 - Delete Record, 11, 280, 321–324, 328–329
 - Display Record Count, 10
 - Dynamic Data, 10
 - Generate Image Dimensions, 412–413
 - Insert Record, 11
 - with multiple-table databases, 310
 - with user registration form, 352
 - wrapping in else clause, 315
 - Log In User, 362–363
 - Log Out User, 365
 - Master Detail Page Set, 11
 - MX Kollection 3
 - pros and cons, 374–375
 - user management server behaviors, 365–370
 - MySQL, required fields, 253
 - overview, 9
 - Recordset, 10
 - Recordset Paging, 10, 276–277
 - Repeated Region, 10, 275–276
 - Restrict Access to Page, 364
 - Show Region, 10, 384
 - display for empty recordsets, 386–387
 - Sticky Edit Field
 - creating, 354–356
 - adding to user update page, 359
 - Sticky Text Area, 348, 355
 - Sticky Text Field, 349–352
 - Stored Procedures, 10
 - third-party products for, 16

- Update Record, 11, 315
 - deleting records using, 280
 - handling insertions, 316–319
 - hidden form fields with, 326–328
 - viewing insertions, 319–320
- User Authentication, 11, 337
 - applying to user registration form, 341–346
- XSL Transformation, 11, 436
- Server Behaviors panel
 - accessing, 261
 - Recordset dialog box, 261–262
- server root
 - designating in Apache, 68
 - directory/folder for, 48
 - test file for, 69
- SERVER (\$_SERVER) superglobal array
 - creating intelligent links using, 403–405
 - uses for, 203
 - viewing options for, 25
 - with basename() function, 203
- server-side includes (SSIs), 122
 - applying styles using, 122–123, 125–126
 - for frequently changed content, 125–126
 - include() function versus require() function, 127–129
 - tips for using, 129
- server-side languages, 3
 - parsing, 47
 - sessions
 - creating, 333
 - destroying, 334
 - function, 332
 - implementing a session, 334–337
 - troubleshoot, 334
 - variables, 333
- server-side technologies, how they work, 3, 7–8
- server-side XSLT
 - advantages, 435
 - checking server support for, 436
 - Dreamweaver handling of, 438
 - RSS news feed display, 437
- servers. *See also* testing server
 - definitions, 4
 - validating forms on, 179–184
- services (Windows), 8
 - running MySQL as, 221
- Services button (Apache Service Monitor), 57
- Services panel (Windows)
 - changing MySQL start method, 227, 229
 - setting Apache startup preferences, 58
- SESSION (\$_SESSION) superglobal array, 333, 335
- sessions
 - creating, 333
 - destroying, 334, 336
 - function of, 332
 - implementation, 334–337
 - security function, 332
 - troubleshooting, 334
 - and User Authentication user behaviors, 337
 - variables, creating and destroying, 333
- session_register()/session_unregister() functions (deprecated), 334
- SET datatype (MySQL), 257
- SET keyword with UPDATE command, 294
- sharing customized workspace layouts, 20–21
- shell prompt (Mac OS X Terminal), 81
- Shift+Enter/Shift+Return keyboard shortcut (new line), 166
- Shift+F9 keyboard shortcut (Snippets panel—Windows only), 177
- Shift+F11 keyboard shortcut (CSS panel—Windows only), 113
- Show as Menu option (Insert bar), 9
- Show as Tabs option (Insert bar), 9
- Show If Recordset Is Empty option (Show Region server behavior), 387
- Show If Recordset Is Not Empty option (Show Region server behavior), 386
- Show If tags with form labels, 188
- Show PHP information (phpMyAdmin), 245
- Show Region server behavior, 10, 384
 - display for empty recordsets, 386–387
 - limits, 188
- showImage() function (JavaScript), 423
- sibling nodes (XML documents), 440
- Simple button (Recordset dialog box), 261
- single quotes (' '), in PHP code, 124
- site definition, 47
 - PHP site, 89–94
 - testing site, 94–95
- Site menu
 - Manage Sites option, 89
 - New Site option, 89–90
- site-root relative links, 48
 - formatting, 49
 - when to use, 49
- Sites folder (Macintosh), 47
- site_check folder, 102
- SMTP settings (Windows systems), 173
- snippets
 - Build image list, 383–384
 - creating and saving, 192
 - editing and deleting, 192
 - Email address check, 180–181
 - Get image dimensions, 411
 - installing, 176
 - POST stripslashes, 177
 - Snippets panel
 - adding files to, 176
 - creating and storing code snippets, 192
 - opening, 177

- sorting. *See also* collation (MySQL)
- recordsets, 263–264
- XML elements in XSLT Fragments, 454
- source documents (XML)
 - formatting in XSLT Fragments, 454–455
 - sorting in XSLT Fragments, 454
- spaces
 - in variable names, 135
 - with MySQL functions, 391
- tag, with form validation code, 183
- Sparber, Al (Project Seven), 117
- Special Characters option (Insert, HTML menu), 109
- spreadsheets versus databases, 252
- SQL (.sql) files, populating MySQL tables using, 258–259
- SQL queries
 - for automatic drop-down menus, 306
 - backticks in, 292
 - combining data, CONCAT() function for, 308–311
 - commands
 - basic syntax, 302
 - case insensitivity, 299
 - DELETE command syntax, 302–303, 320
 - ignoring white space in, 299
 - INSERT command syntax, 301–302
 - ordering, 299
 - punctuation, 299
 - SELECT command syntax, 299
 - UPDATE command syntax, 294, 302
 - WHERE expression with, 303
 - content management using, 312–315
 - creating list of records using, 274–276
 - defined, 213
 - editing, 264
 - keywords, 236
 - moving data between tables, 291–293
 - transferring foreign key, 294–295
 - query process, 10, 297–298
 - SELECT commands
 - arithmetic operators with, 300
 - BETWEEN min AND max with, 300
 - DIV clause with, 301
 - FROM command with, 299
 - IN() function with, 300–301
 - LIMIT clause with, 301
 - logical operators with, 300
 - ORDER BY clause with, 301
 - syntax, 299
 - WHERE clause with, 299–300
 - with LEFT JOIN, 325
 - simple queries, 264
 - sorting recordsets, 263
 - storing recordsets, 264
 - Update Record server behavior with, 320
 - URL parameters with, 401–403
 - viewing in Recordset dialog box, 263
 - SQL tab (phpMyAdmin), Fields list, 291
 - SQLite, 16
 - square brackets ([]), in code hints, 132
 - src attribute, tag, 407–408
 - SSIs. *See* server-side includes
 - Standard Character Set (MySQL), selecting in Windows, 221
 - Standard Configuration option (MySQL Server Instance Config Wizard), 218
 - start options, MySQL
 - Mac OS X systems, 231
 - Windows systems, 227, 229
 - adding PHP to, 65
 - Start screen, DOCTYPE assignment, 37
 - static websites, 2, 6–7
 - Sticky Edit Field server behavior
 - adding to user update page, 359
 - creating, 354–356
 - Sticky Text Field server behavior,
 - adapting for editing, 354–356
 - adding to newuser.php, 349–352
 - creating, 347–348
 - storage capacity, MySQL datatypes, 255–257
 - stored procedures
 - defined, 10
 - not supported in Dreamweaver 8, 10
 - storing
 - data in XML documents, 432
 - dates in MySQL databases, 425–428
 - images in databases, pros and cons, 405
 - recordsets, 264
 - strcmp() function, 352
 - string datatype, 132, 137
 - strings
 - adding text to, 147–148
 - concatenation (joining), 146
 - quotes for, 134, 137
 - striped table rows, creating, 391–392
 - strlen() function with user authentication, 344
 - Structure section (phpMyAdmin Export)
 - Add DROP TABLE option, 471
 - SQL export compatibility option, 471
 - Structured Query Language. *See* SQL queries
 - <style> blocks, attaching stylesheets in, 116
 - style rules (cascading stylesheets, CSS)
 - accommodating Internet Explorer 5 box model, 118–119
 - attaching
 - @import method, 114–116
 - using <link>, 112–113
 - blog site formatting, striped table displays, 391
 - copying
 - Microsoft conditional comment method, 122
 - PHP includes method, 122–123, 125–126

- for feedback form, 167
- for photo gallery display, code for, 416–417
- removing, 117–118
- setting backgrounds using, 392
- for websites
 - adding to dynamic web pages, 267–268
 - changing automatically, 193
 - updating, 196–198
- when to add, 106
- submit button (feedback form), how it works, 168
- sudo command (Mac OS X), resetting root password using, 475
- superglobal arrays
 - security benefits, 171
 - syntax, 170
- switch() control structure
 - structure and syntax, 193–194
 - updating style rules using, 196–198
- syntax
 - DATE_FORMAT() function, 388–389
 - DELETE command, 302
 - INSERT command, 301–302
 - PHP
 - arrays, 168
 - control structures, 134, 151–154
 - double equal signs, 137
 - functions, 134
 - literals, 134
 - variables, 134
 - SELECT commands, 299–300
 - UPDATE command, 302
- Syntax Coloring, enabling, 131
- System Preferences (Mac OS X), accessing configuration files, 79

T

- Tab Index option (Input Tag Accessibility Attributes dialog box), 165
- tabbed documents (Mac OS X), 18–19
- Table dialog box (Insert menu), viewing records in, 273
- Table drop-down menu (Recordset dialog box), 262
- table list (with FROM in SQL queries), 299
- Table-specific privileges (MySQL), 249
- tables (MySQL). *See also* columns (MySQL tables)
 - access privileges, updating, 474–475
 - for blog web site
 - column definitions, 378–379
 - formatting display, 391–392
- datatypes
 - binary data, 257
 - dates and times, 256
 - numbers, 256
 - options, 255

- predefined lists, 257
- primary key options, 253
- text options, 254–255
- displaying XML data in, 448
- multiple
 - backing up, 472
 - creating, 289–291
 - linking using primary/foreign keys, 287
 - revising recordset to handle data from, 297–298
 - transferring data between, 292–293
- MyISAM, foreign key constraint limits, 288
- naming conventions, 292
- for photo gallery
 - creating, 416
 - populating, 417
- populating, 258–259
- primary and foreign key column placement, 290
- revising
 - adding foreign keys, 290–291
 - deleting columns, 295
 - linking with column name in UPDATE queries, 294
 - moving data between, 291–293
 - transferring foreign keys, 294–295
- single
 - backing up, 470–471
 - creating, 252
- for viewing records
 - creating, 273
 - expanding results, 276
 - navigation system for, 276–277
 - query syntax, 274
 - reordering results, 274–275
- viewing in phpMyAdmin, 245
- table types (MySQL)
 - changing default type in Windows, 226–227
 - InnoDB, locating tablespace in Windows, 220
 - MyISAM versus InnoDB, 219–220
- Tag Editor (Code view), 164
 - default Method setting, 305
- Tag Inspector, accessing, 19
- tags
 - <blockquote>, inserting on web page, 265–266
 - in collapsed code, 23
 - <div>, 103–104
 - inserting and identifying, 105–106
 - in XSLT Fragments, 441
 - <form>, in feedback form layout, 163–164
 -
 - alt attribute, 106
 - src attribute, 407–408
 - <input>, 165
 - <label>, 165
 - , in navigation menu automation, 202–203
 - <p>, with SSIs, 125
 - for PHP comments, 139–140

- ``, 183
 - ``, 406
 - `<th>`, 273
 - ``, in navigation menu automation, 203
 - XHTML 1.0 versus HTML 4, 32–33, 35
 - XML
 - attributes, 441
 - custom-made tags, 433–434
 - versus XHTML, 432
 - Tag selector, xvi
 - Tantek box model hack, 111
 - tcsh shell, modifying PATH for MySQL, 233
 - templates
 - alternatives to, 109
 - dynamic web pages as, 5
 - using with PHP, 6
 - versus PHP includes, 125
 - ternary operator (?), 426–428
 - Test Configuration option (Apache), 71
 - Test SQL Statement panel, sorting recordsets in, 262
 - Testing Server category (Site Definition)
 - Access menu, 91
 - importance, 91
 - server folder value, 92
 - Server model menu, 91
 - Testing server folder value, 92
 - URL prefix value, 92–93
 - testing servers
 - local
 - pros and cons, 44–45
 - server root directory/folder, 48
 - setup process, 51–52, 60–69, 71–83, 86–95
 - software needs, 46
 - virtual hosts, 49–50
 - for PHP, configuring in IIS, 76
 - remote
 - testing display capabilities, 45
 - using, 95–96
 - test file/folder, 69, 102
 - troubleshooting connection problems, 260
 - text
 - aligning images with
 - middle of text placements, 408–411
 - top and bottom placements, 406–408
 - displaying, custom server behavior for, 346
 - in MySQL tables, datatype options, 255
 - in PHP code
 - adding to strings, 147–148
 - escape sequences, 146
 - functions for, 396
 - joining strings, 146
 - string datatype for, 132
 - Text Area button (Forms Insert bar), 166
 - Text Field button (Forms Insert bar), 165
 - text fields (feedback forms), labeling, 165–166
 - text length, alternate text, 106
 - text literals, quotes with, 134
 - Text menu, Unordered List option, 108
 - text messages, removing unwanted slashes from, 176–177
 - TextPad script editor, 61
 - TextWrangler
 - file comparison utility, 27–28, 31
 - modifying .profile file, 232–233
 - `<th>` tags, 273
 - thank you message, adding to feedback forms, 186–190
 - third party extensions, for automating contact forms, 162
 - thumbnails (photo gallery displays)
 - activating, 423–424
 - building main image area, 418–422
 - populating gallery table, 417
 - tables and style rules, 416–417
 - tilde (~), in Unix code, 81
 - time (MySQL)
 - datatypes for, 256–257
 - formatting, 388–389
 - TIMESTAMP column, 379
 - TIME argument (DATE_FORMAT() function), 388–390
 - TIMESTAMP datatype (MySQL), 256
 - TINYBLOB datatype (MySQL), 257
 - To email header (feedback form), 175
 - Transactional Database Only (MySQL configuration on Windows), 219
 - trashcan method for removing stylesheets, 117–118
 - true values (Boolean datatype), 137
- ## U
- `` tag, in navigation menu automation, 203
 - underscore (_) character in variable names, 135–136
 - Undo option (Edit menu), 24
 - Unicode (XML documents), 433
 - Unicode Normalization Form document option (Preferences dialog box), 37
 - uninstalling
 - Apache in Windows, 78
 - cautions, 46
 - MySQL
 - Mac OS X systems, 475–478
 - Windows systems, 224
 - PHP in Windows, 78
 - Unix systems, case sensitivity, 236
 - Unordered List option (Text, List menu), 108
 - UNSIGNED datatype (MySQL), 253
 - UPDATE command (SQL)
 - basic syntax, 302
 - Set keyword with, 294
 - transferring foreign key from one table to another using, 294
 - WHERE expression with, 294, 302

- update form
 - adapting for deletes, 321–324, 328–329
 - for blog site, creating, 392–395
- UPDATE privilege (MySQL), 246
- Update Record server behavior, 11
 - advantages of using, 315, 319
 - hidden form fields with, 326–328
- updating
 - databases
 - deleting records, 280–281
 - inserting records, 270–272
 - updating records, 278, 280
 - records, 278, 280
 - cascading updates, 288
 - user accounts, 353, 356–360
- upgrading
 - Apache web server, 52, 77
 - MySQL
 - Mac OS X systems, 475–478
 - updating privileges table following, 474–475
 - Windows systems, 224
 - PHP
 - Mac OS X systems, 80–82
 - Windows systems, 59, 77
- URL Browser (context menu, Code Hint Tools), 124–125
- URL parameters, accessing data using, 401–403
- URL prefix value (Site Definition dialog box), 92–93
- User Administrator section (MySQL Administrator), 240
- User Authentication server behaviors, 11
 - applying to user registration form, 341–346
 - Check New Username, 337, 342–343
 - Log In User, 337, 362–363
 - Log Out User, 337
 - MX Kollektion 3 server behaviors, 367–370
 - Restrict Access to Folder (MX Kollektion 3), 374
 - Restrict Access to Page, 337, 364
- Username field, 248, 260
- User overview screen (phpMyAdmin), 248
- user registration system
 - creating, 337
 - login systems, 362–363, 372–373
 - log out systems, 365, 373
 - MX Kollektion 3 server behaviors for, 365
 - installing, 366–367
 - pros and cons, 374–375
 - Restrict Access to Folder, 374
 - setting up for user authentication, 367–370
 - MX Kollektion User Registration Wizard, 370–372
 - MySQL user accounts
 - creating, 246–248
 - deleting, 250
 - privilege settings, 248–250
 - updating and deleting, 353, 356–361

- user registration system
 - Logout User, 373
 - creating registration form, 339–346, 349–353
 - handling input errors, 352
 - POST stripslashes snippet, 353
 - Sticky Text Field server behavior to, 349–352
 - User Authentication server behaviors, 341–346
 - updating and deleting user accounts, 353, 356–361
 - users table, 338
- users. *See also* user registration systems
 - access restrictions, 364, 374
 - authenticating, 367–370
 - input errors, handling, 349–352
 - PHP request handling process, 7–8
 - validating, 178, 357–358

V

- validation
 - form field checks, 178
 - updating existing user information, 357–358
- values
 - assigning to PHP variables, 136–137
 - passing to and returning from functions, 206–208
- VARCHAR datatype (MySQL)
 - not null with, 253
 - storage capacity, 255
- variables (PHP)
 - assigning values to, 136–137
 - context menus and auto completion for, 25
 - \$ symbol, 135
 - names, 135–136
 - predefined, 170–171
 - in self-processing forms, duration, 185
 - uses for, 134
- version number (PHP), recommended setting, 84
- virtual hosts
 - pros and cons, 49–50
 - registering on Mac OS X, 88
 - registering on Windows, 86–87
 - setup process, 86–89
 - versus virtual directories, 49
- Visual Aids menu (Document toolbar)
 - CSS Layout Backgrounds option, 116
 - CSS Layout Outlines option, 106

W

- warning messages
 - applying to feedback form, 183
 - JavaScript errors, 384
 - PHP errors, 130
 - style rule for, 167
 - when using remote servers, 96

- web pages
 - attaching external JavaScript files, 424
 - binding dynamic data to, 265–266
 - displaying current year and copyright notice, 131–133
 - dynamic content
 - inserting placeholder, 266–267
 - preset formats, 269
 - styling, 267–268
 - viewing, 267
 - PHP includes
 - applying stylesheets using, 122–126
 - changing content using, 125–126
 - include() versus require(), 127–129
 - tips for using, 129
 - selecting using URL browser, 124–125
 - sessions
 - creating, 333
 - destroying, 334
 - function of, 332
 - implementation, 334–337
 - variables, 333
 - static versus dynamic, 6–7
- web servers, 4. *See also* testing servers
 - Apache
 - advantages of using, 52
 - versions, 52
 - Windows installation, 53–56, 58
 - MIME types
 - display problems, 33–34
 - verifying XSLT support, 436
- Web Standards Project (WaSP), 32
- websites
 - connecting to MySQL database
 - MySQL Connection dialog box, 259–260
 - troubleshooting, 260
 - developing and designing, 101, 103–104
 - dynamic data
 - adding style rules, 267–268
 - binding to web pages, 265–267
 - file location, 48–50
 - formatting using preset formats, 269
 - site location, 47
 - viewing, 267
 - virtual hosts for, pros and cons, 49–50
 - Live Data view, 14
 - querying databases, recordset creation, 261–264
 - static versus dynamic, 2
 - style rules
 - updating according to month, 196–198
 - updating automatically, 193
 - uploading, 265
- Welcome screen (phpMyAdmin)
 - Create new database field, 247
 - database settings, 245
 - Privileges link, 248
- well-formed XML documents, 434
- WHERE expression (SQL)
 - adding content using, 325
 - operators used with, 300
 - with DELETE command, 303
 - with SELECT command, 299–300
 - preventing content duplication, 312–315
 - using data from multiple tables, 298
 - with UPDATE command, 294, 302
- while control structure (PHP), 134
- while loops
 - for automating navigation menu, 199
 - code for, 199
- white space
 - in MySQL commands and functions, 299
 - in PHP code, 138
- Windows Server with IIS, 49
- Windows systems
 - Dreamweaver 8 on, 14
 - file comparison utilities, 27
 - MySQL
 - case insensitivity, 236
 - changing default table type, 226–227
 - installing, 214–215
 - removing and upgrading, 224
 - running two series in parallel, 225–226
 - running using MySQL monitor, 229
 - security settings, 222
 - starting and stopping manually, 227, 229
 - updating privileges table, 474
 - MySQL Windows Essentials
 - configuring, 218–221, 223
 - installing, 215–218
 - reconfiguring, 224
 - php.ini with, SMTP settings, 173
 - preset workspace layouts, 17
- Services panel
 - accessing, 58
 - Apache web server startup, 58
- sharing files with Mac systems, 21
- test server setup
 - adding PHP to startup, 65
 - Apache installation, 52–56, 58
 - choosing a web server, 52
 - configuring Apache to work with PHP, 66, 68–69, 71
 - configuring IIS to work with PHP, 73, 75
 - displaying filename extensions, 51
 - enabling filename extensions, 51
 - PHP installation, 58–61, 63–64
 - uninstalling and upgrading
 - Apache, 77–78
 - PHP, 77–78
 - viewing PHP configuration files, 84–85
 - virtual hosts, registering, 86–87

WinMerge file comparison utility, 16, 27
 workspace organization, Dreamweaver 8 enhancements
 for, 17–19, 21
 Wrap around selection (Insert Div Tag), 107

X

XAMMP bundled software package, 46
 Xcode Tools package (OS X), 27
 XHTML (Extensible Hypertext Markup Language), 5
 combining with PHP, 182–184
 DOCTYPEs
 XHTML 1.0 Strict DOCTYPE, 35
 XHTML 1.0 Transitional DOCTYPE, 32, 35
 XHTML 1.1 DOCTYPE, 35
 XHTML 2.0, 33
 XHTML Mobile 1.0 DOCTYPE, 36
 MIME type-related problems, 33–34
 versus HTML 4.0, 32, 35
 versus XML, 432
 XML (Extensible Markup Language)
 custom tags, naming conventions, 433–434
 versus XHTML, 432
 XML declaration/prolog, 433
 XML documents
 attributes, 441
 client-side XSLT, 434–435
 local source documents, 446–447
 remote source documents
 embedding in dynamic pages, 444–446
 linking to XSLT Fragment, 441–443
 structure, 440–441
 server-side XSLT, 435
 Dreamweaver handling of, 438
 RSS newsfeeds, 437
 XSLT Fragment creation and embedding, 439–446
 storing data in, 432–433
 structure, 441
 table displays, 448
 Unicode encoding, 433
 well-formed, 434
 xor keyword, as exclusive OR operator, 156
 XPath, uses for, 442
 XPath Expression Builder
 Build Filter area, 455–457
 building XPath, 442–443
 Format drop-down menu, 454
 XSL Transformation dialog box
 accessing, 444
 adding XSLT Fragments from, 445
 XSL Transformation server behavior, 11, 436

XSLT (Extensible Stylesheet Language Transformations), 16
 adoption as W3C standard, 432
 case sensitivity, 454
 client-side transformations, 434
 limits, 435
 tutorial, 435
 code structure, 448
 conditional regions, 451
 defining new HTML entities, 449
 embedding XHTML in XSLT, 449–450
 filtered output displays, 455–457
 formatting elements, 454–455
 multiple conditional regions, 452–453
 nested repeated elements, 450–451
 parameter-based output displays, 457
 position() and last() functions, 451–452
 sorting elements, 454
 formatting capabilities, 435
 parameters
 displaying output using, 457
 sending from PHP pages, 458–460
 server-side transformations
 advantages, 435
 checking server support for, 436
 Dreamweaver handling of, 438
 RSS news feed display, 437
 using a local XML source, 446–448
 XSLT Fragment creation and embedding, 439–446
 XSLT category (Insert bar), 442
 XSLT Fragments
 code structure, 448
 conditional regions, 451–452
 defining new HTML entities, 449
 embedding XHTML in XSLT, 449–450
 filtered output displays, 455–457
 formatting elements, 454–455
 multiple conditional regions, 452–453
 nested repeated elements, 450–451
 parameter-based output displays, 457
 sending parameters from PHP pages, 458–460
 sorting elements, 454
 creating, 11, 439
 linking to remote XML files, 440–442
 using a local XML source, 446–448
 XPath Expression Builder, 442–443
 embedding in dynamic pages, 444–446
 function, 448

Y

year, current, displaying on web pages, 131–133