**The Storefront Main Section**

The main section in your storefront should provide information that you want the customer to see immediately. You can include a greeting, store notices, and special event items, such as sale items or news flashes.

The Food Store storefront main section displays a simple greeting, followed by the products that are currently on sale. This allows customers to select sale items easily without having to hunt for them in the normal catalog listings.

To do this, you'll need to query the products database to find the items that the manager marked as being on sale. Remember, the *onsale* data field in the products table indicates if a product is on sale. It's a Boolean value, so you'll need to be careful when querying for this value. There's a simple trick to create the query for this:

SELECT \* FROM products WHERE onsale = TRUE

The Boolean data type has two possible values—0 or 1. When you perform your query, you can use the numerical values of zero or one, or you can use alias values of TRUE (for one) or FALSE (for zero). In the application, a TRUE value indicates the product is on sale.

When you display the list of products that are on sale, it's a good idea to provide as much information as possible (including the stored image), and a link, so the customer can purchase the product directly from the main Web page.

Let's build the main.inc.php file, which displays the content for the main section:

1. Create the file *main.inc.php* in the store folder.
2. Open the file in an editor, and enter the following code:

<h2>Welcome to our store!</h2>

<br>

<br>

<p>Please feel free to browse our great selection of products. Select the category

from the drop-down menu in the navigation bar. Add items to your cart, then check out.

<p>

<h2>Items on sale today:</h2>

<?php

$query = "SELECT \* from products where onsale = TRUE";

$result = mysql\_query($query);

echo "<table width=\"100%\" border=\"0\">\n";

while($row=mysql\_fetch\_array($result, MYSQL\_ASSOC))

{

$prodid = $row['prodid'];

$description = $row['description'];

$price = $row['price'];

$quantity = $row['quantity'];

echo "<tr><td>\n";

echo "<img src=\"showimage.php?id=$prodid\" width=\"80\" height=\"60\">";

echo "</td><td>\n";

if ($quantity == 0)

echo "<font size=\"3\">$description</font>\n";

else

{

echo "<a href=\"index.php?&content;=updatecart&id;=$prodid\">";

echo "<font size=\"3\"><b><u>$description</u></b></font>\n";

}

echo "</td><td>\n";

printf("$%.2f\n", $price);

echo "</td><td>\n";

if ($quantity == 0)

echo "<font color=\"red\">Sorry, this item out of stock</font>\n";

else if ($quantity < 5)

echo "Hurry, only $quantity left in stock!\n";

else

echo " \n";

echo "</td></tr>\n";

}

echo "</table>\n";

?>

1. Save the file and exit the editor.
2. Notice that this code uses the showimage.php file you created earlier for the back-end application, so you could display an image on a Web page. Copy the *showimage.php* file from the admin folder to the store folder.

The main.inc.php code displays a short message about the store to the customer. It then performs a simple SQL query, looking for all the products marked as on sale. It iterates through the returned data records, displaying each product on a separate line.

You'll note that the code checks the quantity in stock for each product. If a product has more than five in stock, no special messages are displayed, and the product description is displayed as a link to the *updatecart* content (which we'll discuss later). If a product has less than five in stock, the code displays a special message indicating how many are left. Finally, if a product is out of stock, the code displays the product description as regular text without the link (since there aren't any to purchase).

You may have noticed that I snuck in a different way of displaying the price data field:

printf("$%.2f\n", $price);

If you've ever used the C programming language, you know exactly what the printf() function does. If you're not a C programming guru, let me explain a little.

The *printf()* function provides for extremely fine control over displaying data. It uses special formatting codes that allow you to define what type of data to display and how to display it. The format of the printf() function is:

printf("format", variable list)

The *format* parameter defines what you want the printf() function to print. The format parameter consists of normal text, along with special codes for printing variables. There are lots of special codes that the printf() function supports, but the ones most often used are:

|  |  |
| --- | --- |
| **The printf() function codes** | |
| **Code** | **Description** |
| %c | Displays a single character value |
| %d | Displays an integer value |
| %f | Displays a floating point value |
| %s | Displays a string value |

The format string contains special codes for each variable you want to display. You must then specify the variables in the correct order in the *variables list*. This goes something like this:

printf("You purchased %d of %s", $quantity, $description);

The printf() function matches the $quantity variable value with the %d code to display the integer value, and the $description variable value with the %s code to display the string value.

Now here's where it gets interesting. You defined the price data field in the products table as a decimal(6,2) data type. When you extract this value, using the mysql\_fetch\_array() function, it becomes a floating point value in PHP. If you use the echo statement to display this value, it'll drop the decimal places if they're zero. This would look somewhat clunky for customers. They're used to seeing dollar values look like $1.50 and not $1.5.

The printf() function allows you to not only specify that a data element is a floating point value with the %f code, it also allows you to specify the number of decimal places to always display. You do that by placing the value between the percent sign and the f code:

printf("The cost of the item is $%.2f\n", $price);

This format will always display a two-decimal place value, even if one or both of the trailing values is zero.

This demonstrates the power you have while generating Web sites with PHP code. You can display your data in any format you desire by creating the HTML code within your PHP printf() statements.

Let's continue on to Chapter 4 and talk about the last section on the storefront home page, the shopping cart.