



Quick answers to common problems

OpenCart 1.4 Template Design Cookbook

Over 50 incredibly effective and quick recipes for building modern eye-catching OpenCart templates

Tahsin Hasan

[PACKT] open source 
PUBLISHING community experience distilled

OpenCart 1.4 Template Design Cookbook

Over 50 incredibly effective and quick recipes for building modern eye-catching OpenCart templates

Tahsin Hasan

[PACKT] open source 
PUBLISHING community experience distilled

BIRMINGHAM - MUMBAI

OpenCart 1.4 Template Design Cookbook

Copyright © 2011 Packt Publishing

All rights reserved. No part of this book may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, without the prior written permission of the publisher, except in the case of brief quotations embedded in critical articles or reviews.

Every effort has been made in the preparation of this book to ensure the accuracy of the information presented. However, the information contained in this book is sold without warranty, either express or implied. Neither the author, nor Packt Publishing, and its dealers and distributors will be held liable for any damages caused or alleged to be caused directly or indirectly by this book.

Packt Publishing has endeavored to provide trademark information about all of the companies and products mentioned in this book by the appropriate use of capitals. However, Packt Publishing cannot guarantee the accuracy of this information.

First published: March 2011

Production Reference: 1150311

Published by Packt Publishing Ltd.
32 Lincoln Road
Olton
Birmingham, B27 6PA, UK.

ISBN 978-1-849514-30-9

www.packtpub.com

Cover Image by Asher Wishkerman (a.wishkerman@mpic.de)

Credits

Author

Tahsin Hasan

Editorial Team Leader

Aanchal Kumar

Reviewers

Jimit Modi

Michael Blanco

Nikita Chudjakov

Tomáš Kissik

Project Team Leader

Priya Mukherji

Project Coordinator

Jovita Pinto

Acquisition Editor

Steven Wilding

Proofreader

Mario Cecere

Development Editor

Alina Lewis

Graphics

Nilesh R. Mohite

Technical Editor

Ajay Shanker

Production Coordinator

Aparna Bhagat

Indexers

Monica Ajmera

Tejal Daruwala

Cover Work

Aparna Bhagat

About the Author

Tahsin Hasan is a software engineer from Bangladesh. He passed the Zend Certification Examination on 9th August, 2009 and became the seventeenth Zend Certified Engineer (ZCE) from Bangladesh. This is the top-most certification on PHP from Zend, the developer of this outstanding scripting language. He is a tech enthusiast and always keeps himself well-equipped with latest technologies. He has completed his M.Sc. and B.Sc. in Computer Science and Engineering from Jahangirnagar University, Bangladesh.

Tahsin Hasan has profound knowledge of LAMP environment. His advanced understanding of database environments and Apache web server is an asset. He has proficiency in scalability and optimizing server performance. He has worked with Symfony, CakePHP, Codeigniter, and Zend Framework.

Tahsin Hasan shares his ideas and knowledge on tahSin's gaRage; the address is <http://newdailyblog.blogspot.com>. He welcomes everyone to his blog to discuss latest web technologies.

First of all, I like to thank the Almighty Allah. I also give thanks to my family members for their support.

I would like to thank to Packt Publishing for giving me the opportunity to share my knowledge on this excellent topic. Especially, I like to thank Steven Wilding and Jovita Pinto for their efforts. Also, a special thanks goes to the reviewers.

Most specially, I thank my readers for their eagerness to read the book.

About the Reviewers

Michael Blanco has over 4 years experience in the IT industry. He has a passion for Web and software development, which was molded when he was still studying for his Bachelor's Degree In Information Technology at one of the best IT schools in the Philippines which is the University Of San Jose-Recoletos. During his senior college year in 2005, he participated as one of the top ten finalists in a nationwide programming contest held by Trend Micro which is the 2005 Trend Micro Tech Challenge. He acted as a Team Leader and led a group of six IT and Computer Science students to develop their own anti-spyware/anti-malware scanner that detects and prevents a set of sample spywares given by Trend Micro.

He is currently working as a full time Web Developer at BO Technology, LLC, a New York-based French IT Services company that specializes in Web and Ecommerce development specifically in Magento and Virtuemart.

During the past four years, he has developed several tailored business software and websites from various small companies in the Philippines. Up to this day, he still develops Ecommerce websites and software applications on weekends. He says that it's a hobby to create and develop websites or software as an art while generating extra income from it.

He got interested in OpenCart and has developed several free popular extensions for it, which are available for free download in the OpenCart website.

He is also quoted as saying that "Making websites or software is like creating kites for an enthusiastic kid who has the passion not only to fly kites but also to take care of his/her kite which was specially crafted by me. I love clients who value the website/software that I made for them as if it is a masterpiece."

I would like to thank my family especially my mother, Conching for bringing me in to this world, my friends and colleagues at work, and thank you to Packt Publishing for selecting me as one of the technical reviewers in this wonderful book. It is my honor to be a part of this book.

Nikita Chudjakov is a freelance web-developer and an enthusiastic consultant with the mission to help NGOs and small businesses to fully embrace modern open-source solutions in a pragmatic and productive fashion, providing advice, training, encouragement, and customized solutions.

Born in Belarus but currently living in Ghent, Belgium, Nikita has been fascinated by computers since 1989, starting with Basic and beautiful games on Yamaha MSX, followed by Turbo Pascal, dBase, Foxpro on PC, his first experiments with the Web and HTML in the second half of the 90s and his present passion for everything community-created and open-source (especially LAMP, Drupal, CodeIgniter, and OpenCart).

You can contact Nikita via his LinkedIn page: <http://www.linkedin.com/in/nikitachudjakov>

I would like to thank my wife Sveta for her love, support, and her patience with my frequent all-nighters, my parents for their care and upbringing, friends for being friends and all those people willing to donate their time, effort, and expertise by participating in open source software projects, making information available to everyone, fighting for the environmental and relief causes and in any other way helping to make our world a better place.

Tomáš Kissík is a citizen of the Slovak Republic. He studies finance at the University of Economics, Prague, in the Czech Republic. He has been working with Web technologies, PHP and CMS systems for five years. This book is the second title he has worked on after OpenCart Beginners guide.

I want to thank to my family, friends, and people from the OpenCart, community for their support.

www.PacktPub.com

Support files, eBooks, discount offers, and more

You might want to visit www.PacktPub.com for support files and downloads related to your book.

Did you know that Packt offers eBook versions of every book published, with PDF and ePub files available? You can upgrade to the eBook version at www.PacktPub.com and as a print book customer, you are entitled to a discount on the eBook copy. Get in touch with us at service@packtpub.com for more details.

At www.PacktPub.com, you can also read a collection of free technical articles, sign up for a range of free newsletters and receive exclusive discounts and offers on Packt books and eBooks.



<http://PacktLib.PacktPub.com>

Do you need instant solutions to your IT questions? PacktLib is Packt's online digital book library. Here, you can access, read and search across Packt's entire library of books.

Why Subscribe?

- Fully searchable across every book published by Packt
- Copy & paste, print and bookmark content
- On demand and accessible via web browser

Free Access for Packt account holders

If you have an account with Packt at www.PacktPub.com, you can use this to access PacktLib today and view nine entirely free books. Simply use your login credentials for immediate access.

Table of Contents

Preface	1
Chapter 1: Getting Started	5
Introduction	5
Why do we need to customize our theme?	6
Setting up our environment	6
Installing OpenCart	10
Downloading and installing GIMP	14
Creating logos in OpenCart	14
Inspecting elements using Firebug	25
Installing Web Developer	26
Installing Colorzilla	28
Chapter 2: Store Decoration	33
Introduction	33
Changing your shop's general settings	35
Modifying the settings of your store	37
Basic localization settings	40
Setting image properties	45
Changing administrator favorites icon	49
Setting up SMTP mail server	51
Modifying server options	53
Creating categories	54
Adding products to a category	58
Chapter 3: Layout Structure	61
Introduction	61
Creating a new theme	63
Resetting layout styles	64
Setting basic style properties	68

Creating a promotional banner	75
Placing a banner on your site	84
Chapter 4: Module Adjustment	87
Introduction	87
Changing header of module	90
Adjusting layout	94
Placing featured block at middle	98
Styling centre featured block	100
Applying styles to categories	110
Chapter 5: Header Structure	117
Introduction	117
Default header structure	118
Altering the logo	118
Styling top navigation	119
Placing menus	122
Styling the search bar	124
Styling the language bar	128
Controlling search style	133
Styling language and payment units	135
Creating the header banner	138
Chapter 6: Dynamic Content	143
Introduction	143
Installing jQuery and jCarousel	144
Displaying products using jCarousel	145
Displaying products vertically	153
Customizing the cycle plugin	160
Modifying the side column with the cycle plugin	163
Different effects with the cycle plugin	166
Chapter 7: Customizing Menus	175
Introduction	175
Displaying horizontal sliding menus	175
Adding a fisheye menu	187
Giving a background color to the menu	196
Integrating the social networking menu	202
Chapter 8: Footer Layout	207
Introduction	207
Creating a wide footer for our store	207
Footer with a gradient image	216
Creating a three column footer	227

Chapter 9: Modifying the Administration Panel Theme	241
Introduction	241
Modifying headers	241
Menu creation	244
Footer adjustment	248
Styling dashboard	260
Chapter 10: Miscellaneous	281
Introduction	281
Broken glass 404 error page	281
Sitemap creation	295
Index	305

Preface

Templates in OpenCart provide a powerful way to make your site look exactly the way you want either using a single template for the entire site or a separate template for each site section. Although it sounds like an easy task to build and maintain templates, it can be challenging to get beyond the basics and customize templates to meet your needs perfectly.

What this book covers

Chapter 1, Getting Started, introduces the reader to OpenCart, and helps set up the required environment for the template design; where the material in this chapter alone is not enough, the user is pointed to other resources to fill in the gaps of knowledge to proceed further.

Chapter 2, Store Decoration, introduces to the reader the ways of store setting management. We set different basic settings for our store decoration and development. We also create a favicon for our shop and upload products to different categories.

Chapter 3, Layout Structure, discusses the layout structure of OpenCart. Also shown are the steps to create a new theme for OpenCart and how to reset the browser's default styles. This chapter covers styling properties and banner creation for the store, and how it can be implemented in OpenCart.

Chapter 4, Module Adjustment, covers how modules can be adjusted on an OpenCart website using content elements. It also covers how to embed styles and images in different modules. You will learn to render styles to featured blocks and category blocks of an OpenCart store.

Chapter 5, Header Structure, explains how you will modify the header of your store. You will learn the default header structure of OpenCart and will see ways to adjust that structure according to your need. We will also create new banners and menu styles in this chapter.

Chapter 6, Dynamic Content, shows how we can add different jQuery plugins to expand the system. We display our products with different styles and they will appear in a modern attractive way. We also discuss about different styling effects of jQuery plugins that we can use to enhance the look and feel of our store.

Chapter 7, Customizing Menus, covers creation of attractive menus in OpenCart, which allows individual sites to add different types of menus depending on their choice. We have shown different ways to create menus and different stylish images for them.

Chapter 8, Footer Layout, covers how we can create a wide area footer for our store; we will also create a stylish footer with different images in this chapter. We'll also see the creation of a three column footer in this chapter.

Chapter 9, Modifying the Administration Panel Theme, shows how we can add different styles to expand the administration system. We display our products with different styles and they will appear in a modern, attractive way. We also discuss about header and footer styling that we can use to enhance the look and feel of our store.

Chapter 10, Miscellaneous, covers creation of an attractive 404 page and sitemap for our store. We use different styling effects for those pages that we can use to enhance the look and feel of our store.

What you need for this book

In order to get the most from this book, there are some expectations of prior knowledge and experience. It is assumed that the reader has some understanding of HTML, CSS, which can be achieved by reading the introductory tutorials.

Who this book is for

This book is for anyone who is looking for effective ways of managing and operating stores with different content. You will find this book interesting if you are running, or starting Ecommerce websites rich in modern content. This book is very easy to understand. The images are descriptive, so you will learn pretty easily with the images.

Conventions


In this book, you will find a number of styles of text that distinguish between different kinds of information. Here are some examples of these styles, and an explanation of their meaning.


Code words in text are shown as follows: "We still need to set the attribute `cell-spacing` to 0."

A block of code is set as follows:

```
table {  
    border-collapse: collapse;  
    border-spacing: 0;  
}
```

New terms and **important words** are shown in bold. Words that you see on the screen, in menus or dialog boxes for example, appear in the text like this: "clicking the **Next** button moves you to the next screen".

 Warnings or important notes appear in a box like this.

 Tips and tricks appear like this.

Reader feedback

Feedback from our readers is always welcome. Let us know what you think about this book—what you liked or may have disliked. Reader feedback is important for us to develop titles that you really get the most out of.


To send us general feedback, simply send an e-mail to feedback@packtpub.com, and mention the book title via the subject of your message.

If there is a book that you need and would like to see us publish, please send us a note in the **SUGGEST A TITLE** form on www.packtpub.com or e-mail suggest@packtpub.com.

If there is a topic that you have expertise in and you are interested in either writing or contributing to a book on, see our author guide on www.packtpub.com/authors.

Customer support

Now that you are the proud owner of a Packt book, we have a number of things to help you to get the most from your purchase.

 **Downloading the example code for this book**
You can download the example code files for all Packt books you have purchased from your account at <http://www.PacktPub.com>. If you purchased this book elsewhere, you can visit <http://www.PacktPub.com/support> and register to have the files e-mailed directly to you.

Errata

Although we have taken every care to ensure the accuracy of our content, mistakes do happen. If you find a mistake in one of our books—maybe a mistake in the text or the code—we would be grateful if you would report this to us. By doing so, you can save other readers from frustration and help us improve subsequent versions of this book. If you find any errata, please report them by visiting <http://www.packtpub.com/support>, selecting your book, clicking on the **errata submission form** link, and entering the details of your errata. Once your errata are verified, your submission will be accepted and the errata will be uploaded on our website, or added to any list of existing errata, under the Errata section of that title. Any existing errata can be viewed by selecting your title from <http://www.packtpub.com/support>.

Piracy

Piracy of copyright material on the Internet is an ongoing problem across all media. At Packt, we take the protection of our copyright and licenses very seriously. If you come across any illegal copies of our works, in any form, on the Internet, please provide us with the location address or website name immediately so that we can pursue a remedy.

Please contact us at copyright@packtpub.com with a link to the suspected pirated material.

We appreciate your help in protecting our authors, and our ability to bring you valuable content.

Questions

You can contact us at questions@packtpub.com if you are having a problem with any aspect of the book, and we will do our best to address it.

1

Getting Started

In this chapter, we will cover:

- ▶ Setting up our environment
- ▶ Installing OpenCart
- ▶ Downloading and installing GIMP
- ▶ Creating logos in OpenCart
- ▶ Inspecting elements using Firebug
- ▶ Installing Web Developer
- ▶ Installing Colorzilla
- ▶ Latest designing trends

Introduction

OpenCart is an ecommerce solution based on MVC structure. MVC means model, view, controller. OpenCart's structure is easier to follow and understand compared to other ecommerce solutions. That's why, its popularity is increasing rapidly. This chapter will help us to start customizing OpenCart's theme properly.

In this chapter, we will focus on What OpenCart is, its installation and configuration, and the importance of theme designing and several different tools for theme designing.

As the name specifies, OpenCart is an open source ecommerce solution. It is developed with MVC design pattern. OpenCart is a robust e-commerce solution for Internet merchants with the ability to create their own online business. Daniel Kerr is the creator of OpenCart. OpenCart's code is streamlined and easier to follow than some other bloated open source ecommerce solutions such as OsCommerce and Zencart.

Currently, OpenCart supports 20+ payment gateways and 8+ shipping methods.modules. OpenCart is gaining popularity day by day.

OpenCart is an open source PHP-based online Shopping Cart System. It comes with several different kinds of features. OpenCart's template system is pretty easy to get grips with and an original template can be created with some HTML/CSS and image creation skills. However, if one does not possess these skills, free templates can be found on numerous places on the Web including the OpenCart contribs page. Templates are easy to install and we can install our own logo through the OpenCart admin area.

The current version is 1.4.9. The current release boasts a multitude of great features, including multi-language capabilities, product ratings and downloadable products, and product review capabilities. We can also write reviews in the admin panel! Awesome! There is built-in compatibility with a number of payment gateways such as Worldpay and Sagepay. PayPal is supported too and Google Checkout will be in the upcoming releases along with: Google Base, Google Talk, and Google Site Map. Let's see the features of OpenCart:

- ▶ OpenCart is open source
- ▶ It is also module based
- ▶ Templatable system
- ▶ OpenCart supports downloadable products
- ▶ OpenCart brings product ratings and reviews
- ▶ It follows search engine optimization
- ▶ It also supports automatic image resizing features
- ▶ There is support for multiple payment gateways and shipping modules
- ▶ And many more...

Why do we need to customize our theme?

OpenCart comes up with a default theme, and also there are a lot of free themes available for OpenCart. So, why do we need to customize our theme? We can use the ones that are freely available. But, the freely available themes do not match our designing style. Often, they lack modern designing techniques and layouts. Some people prefer to show a single-column page, some double-column, and some a triple-column layout. So, to attract users to our site, we need our design to be attractive and modern.

Setting up our environment

We have discussed a lot, now let's start working. Our first recipe is an environment setup. Without the proper environment, it is difficult to work. Let's first know what the requirements are to work with OpenCart.

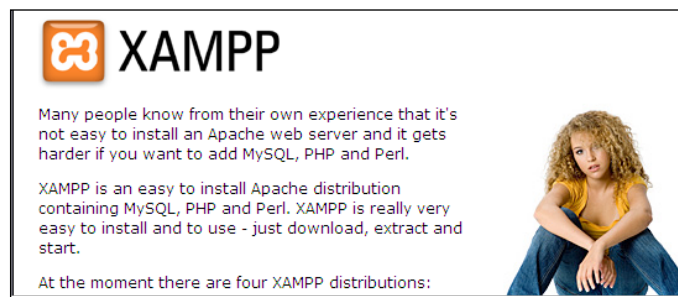
Getting ready

OpenCart works with at least PHP 5. So, we must ensure at least the support of PHP 5. OK, let's see what other things we need:

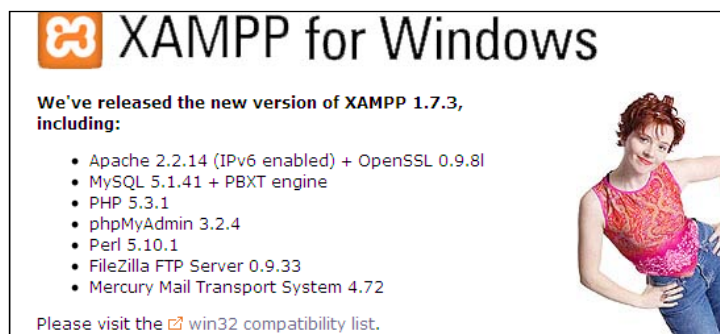
- ▶ PHP 5
- ▶ MySQL
- ▶ Web server (preferably Apache)

We are going to use Xampp for setting up the environment. Xampp is a free and open source cross-platform, web server package. It consists mainly of the Apache HTTP Server, MySQL database, and interpreters for scripts written in the PHP and Perl programming languages. The 'X' in Xampp means cross platform. So, Xampp is available for Microsoft Windows, Linux, Solaris, and Mac OS X, and is mainly used for web development projects.

There are different Xampp packages for different operating systems, so, grab the one you need for your specific operating system. Xampp requires only one .zip, .tar, or .exe file to be downloaded and run, and little or no configuration of the various components that make up the web server is required. Here is the Xampp website link for download and installation process: <http://www.apachefriends.org/en/xampp.html>.



This site has good documentation about the installation process of Xampp for different operating systems. We will see the installation process for Windows here. Go to this link: <http://www.apachefriends.org/en/xampp-windows.html>.



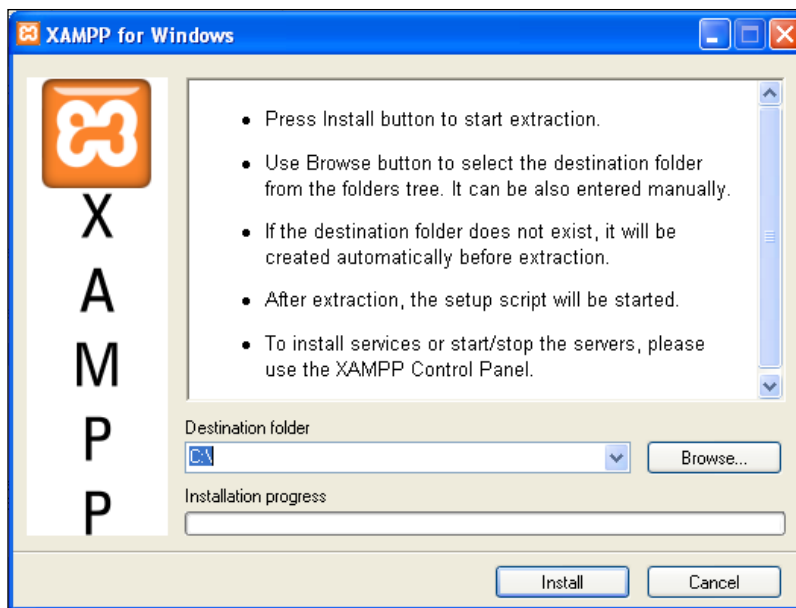
We will be using Xampp 1.7.3. It comprises the following:

- ▶ Apache 2.2.14
- ▶ MySQL 5.1.41
- ▶ PHP 5.3.1
- ▶ phpMyAdmin 3.2.4
- ▶ perl 5.10.1
- ▶ FileZilla 0.9.33
- ▶ Mercury 4.72

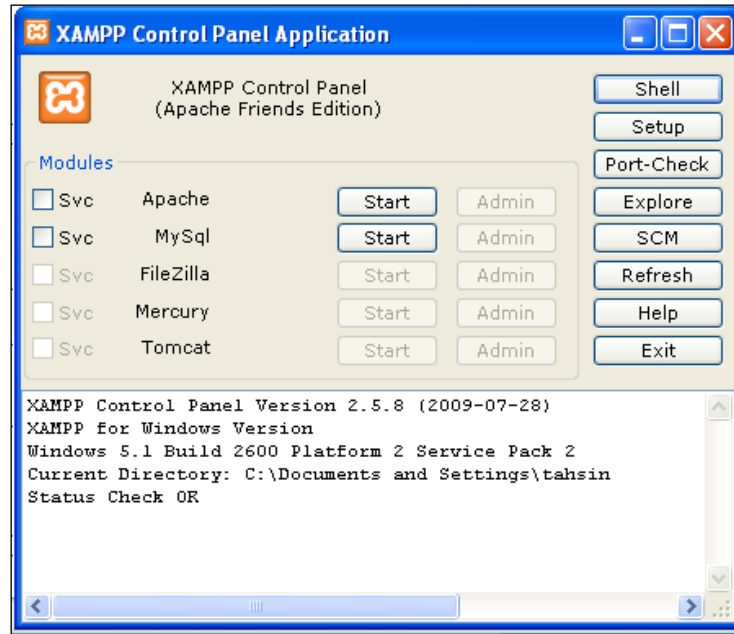
How to do it...

OK, now get the .exe file that is 51.2 MB. We will install Xampp in a drive other than the C drive, because if we reinstall Windows in our machine, then our files will be deleted. Say we install it in the E drive. Follow the steps to install Xampp:

1. Double click the installer,
2. Choose the appropriate folder for installation.



3. Follow the instructions on the installation window. This window will show installation progress status.
4. Now, open the Xampp control panel. There, we will see separate controls for Apache, MySQL, FileZilla, and Mercury.
5. Click the **Start** button to start the respective features. There is a status message of the current state of the selected feature.



We have installed Xampp successfully.

There's more...

There are other packages of AMP families. We have discussed only Xampp, as we have seen earlier that it is cross platform. There are also other web server packages:

1. Wamp for Windows environment
2. Lamp for Linux environment
3. Mamp for Mac environment

Installing OpenCart

Let's get our OpenCart package from the Web. We can find it on their website:
<http://www.opencart.com/index.php?route=download/download>.



We will download the latest version. For this book, we will be using OpenCart version 1.4.8.

Downloads		Changelog
Version	Release Date	
v1.4.8b	June 14, 2010	Download
v1.4.8	June 9, 2010	Download
v1.4.7	April 18, 2010	Download
v1.4.6	April 8, 2010	Download

How to do it...

We have downloaded our OpenCart package. Now, we will extract it and copy the downloaded folder of the OpenCart package into the `htdocs` folder of Xampp and rename the folder to our project name. Let's, name it 'shop'. We need to ensure read and write permissions for the following files and folders:

```
image/  
image/cache/  
cache/  
download/  
config.php  
admin/config.php
```

Database creation

We need to create a database for our OpenCart project. We will go to our phpMyAdmin using the URL: <http://localhost/phpmyadmin>. We are going to create our database. Let's call our database `open_book` with collation `utf8_general_ci`. This collation will improve database searching. Write the database name here for future reference:

Database Name: _____.

We will create a new user for our database with limited user privileges. This will increase database security. For this, go to `privileges->add_a_new_user`. Write down the following information that we set in the new user window, here, for future use:

User Name:_____.

Host Name:_____.

Password:_____.

Setting up virtual host

We are going to set up virtual host for our shop. First, let's see how to set it up in the Linux environment. We will create a `vhost` file under `/etc/sites-available/`.

Then, type the following command in the terminal:

```
a2ensite project_name
```

Here, `project_name` will be our folder name under the document root folder. Now, we will write an extra following line in the `hosts` file under `/etc/` folder.

Now, restart Apache by typing the following command in the terminal:

```
/etc/init.d/apache2 reload
```

Access your browser and type the project URL, we will see our site.

The setup of the virtual host is different in the Windows environment than Linux. From Apache 2.2 onwards, we have a different `vhost` file to create the virtual host under the `xampp/apache/conf/extra` folder named `httpd-vhosts.conf`. Open the file; following is the code block that we need to write for each of our projects:

```
<VirtualHost *:80>
    ServerAdmin webmaster@shop.com
    DocumentRoot D:\xampp\htdocs\shop
    ServerName local.shop.com
    ErrorLog E:\error\local.shop.com-error_log
    CustomLog E:\error\local.shop.com-access_log common
</VirtualHost>
```

OpenCart configuration settings

We have set up our virtual host. Now, type the project URL: `http://localhost/shop` on the browser.

- **Step-01:** First, we will see the licence agreement option of OpenCart. Read the licence and mark the agree option of the licence agreement and then click on the **continue** button.

- Step-02: This shows the necessary pre-installation configuration setup of our server. There is a green tick mark, which is OK, and there is a red mark that needs to be corrected. It also shows the writable options. If everything is in the correct state, click on the **continue** button.

Step 2 - Pre-Installation

1. Please configure your PHP settings to match requirements listed below.

PHP Settings	Current Settings	Required Settings	Status
PHP Version:	5.3.0	5.0+	✓
Register Globals:	Off	Off	✓
Magic Quotes GPC:	Off	Off	✓
File Uploads:	On	On	✓
Session Auto Start:	Off	Off	✓

2. Please make sure the extensions listed below are installed.

Extension	Current Settings	Required Settings	Status
MySQL:	On	On	✓
GD:	On	On	✓
cURL:	Off	On	✗

- License
- **Pre-Installation**
- Configuration
- Finished

- Step-03: This shows the configuration options that we need to set for our project. They are database configuration settings:

Step 3 - Configuration

1. Please enter your database connection details.

* Database Host:

* User:

Password:

* Database Name:

Database Prefix:


Let's discuss the fields:

- ❑ **Database Host:** Previously, we wrote down our database host name for future use. We will enter our database host name here.
- ❑ **Database User:** This will be our database user. We also stored this previously.

- ❑ **Database Password:** We will enter our database password.
- ❑ **Database name:** This is our database name.

2. Please enter a username and password for the administration.

* Username:	<input type="text" value="admin"/>
* Password:	<input type="password"/>
* E-Mail:	<input type="text"/>



Let's discuss the administration fields here:

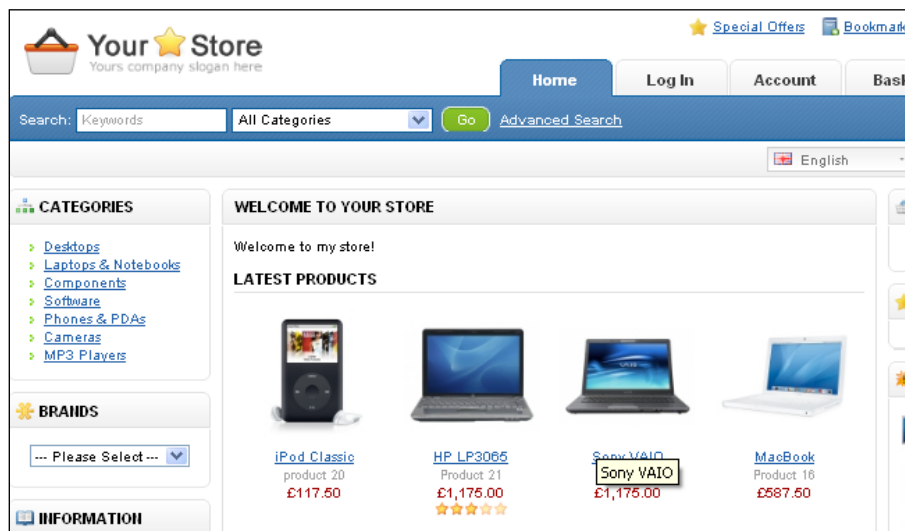
- ❑ **Username:** This will be our administration username. We will write it down here for future reference.
Admin Username: _____.
- ❑ **Password:** This will be our password for admin panel. We will store it here for future reference.
Password: _____.

Running the site

We have everything configured; the stage is set for viewing our OpenCart project default site. Open your favourite browser, then type our project URL:

`http://local.shop.com/`

Our default site will appear like the following:



Post install configuration

If for any reason we need to modify the settings that were entered during the installation, there are two configuration files that can be modified. Settings applied to the catalog area: `<install dir>/config.php`

Settings applied to the administration area: `<install dir>/admin/config.php`

We have set up our OpenCart project 'shop'. Now, let's discuss the tools that we are going to use throughout this book.

Downloading and installing GIMP

GIMP is an open source graphics software editor. It stands for GNU Image Manipulation Program. It is primarily employed as an image retouching and editing tool. GIMP is becoming a high-end graphics application for the editing and creation of original images, icons, graphical elements of web pages, and art for user interface elements.

GIMP was originally created for UNIX systems. But now, it supports Windows and Mac OSX.

GIMP will be our main designing tool. We will see how to create different kinds of images with it using various effects. GIMP is open source and it can be used both in Windows and Linux environments. GIMP has great varieties of add-ons to use.

Getting ready

For Linux, GIMP is added to the accessories by default. So, no hassle for Linux users. For Windows, to download GIMP, we will go to <http://www.gimp.org/downloads/>. We can get both Windows and Linux installers from this site.

How to do it ...

1. We have downloaded GIMP. Now, double click on the installer.
2. We will just click on the **next** button; it shows the license agreement page. It shows the GNU general public license agreement for GIMP.
3. By clicking **next**, we move to the destination selection page. We will set where we want to install our GIMP, it can be `E:\install\GIMP`. In this window, there is a message stating that we need 50 MB of free space for installing GIMP. Click **next**.

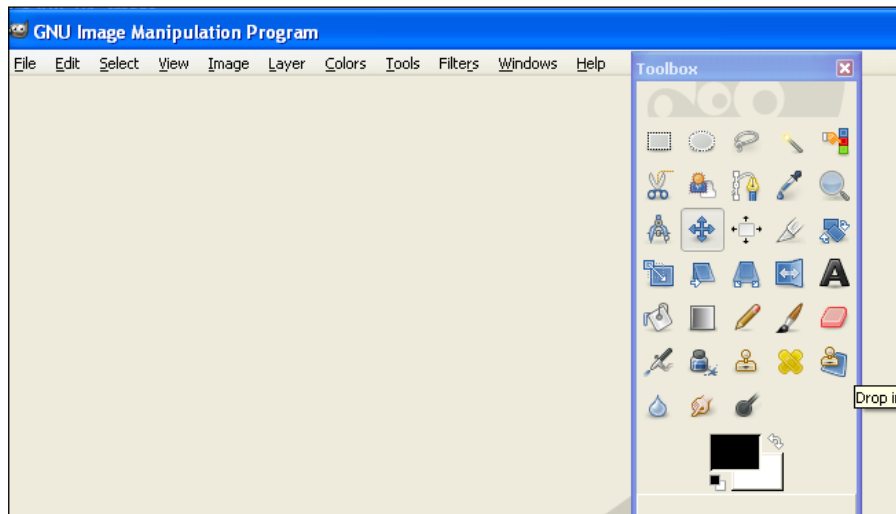
Creating logos in OpenCart

Now, we will create a logo for our store. Our logo will be an identity of our store. We will create a simple logo for our project 'shops'. As this is our first time working with GIMP, we will see a step-by-step guide for creating a logo with GIMP.

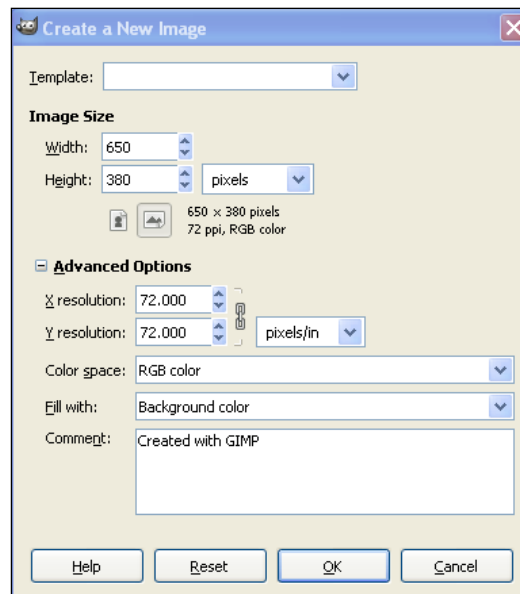
How to do it...

1. We need to open GIMP. Go to **Start | All Programs | GIMP | GIMP 2** or **Application | Graphics | GIMP**.

The default window of GIMP is shown.



2. To create a new image, we will go to **File | New** of GIMP. We can use **Ctrl+N** as the shortcut option for creating a new image. The following window will be open.

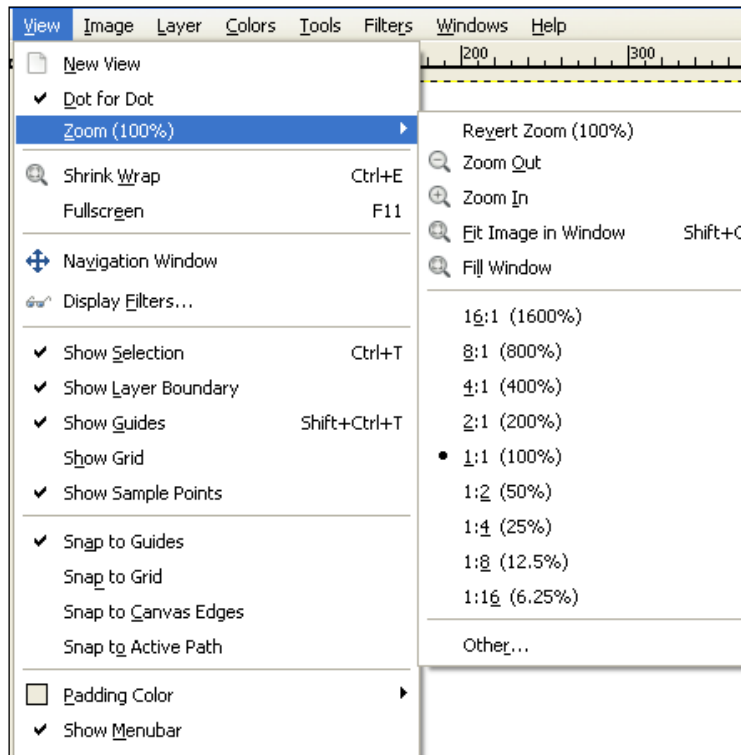


Let's discuss every option:

- ❑ **Template:** There is a drop-down list for some defined templates. These templates are some of the common useful images that we frequently need for our development. You can see some defined height and width are given in this option. By choosing this option, we can set the height and width of our image. If we choose an option from the drop down, then the height and width section below the template will be populated automatically. We can also leave this option blank and can set height and width of our images manually. In our case, we will leave this blank.
- ❑ **Image size:** Here, we can set the height and width of the images. The sizes can be set in different units. In the height and width textbox, we can type in the values, and we can also adjust those values with the up and down arrows in the respective textboxes. Our logo will have a height of 43px and a width of 210px.
- ❑ **Units:** There are many units available in GIMP for image creation. We can select our chosen units from the drop down. You can also see a **More...** link below the drop down. There, we can find more unit options that are available for our use.
- ❑ **Portrait/Landscape:** Under the size section, there is an option to select for portrait/landscape. When portrait is selected, the height is larger than the width. If we select landscape, the width will be greater than the height.
- ❑ **X resolution:** We can set the X resolution for our image here with the proper unit. We will leave this as it is. The link icon lies besides the x and y resolution. When the link is connected, and you make a change in one textbox, the other will automatically change. If we unlink it, then making a change in one will not affect the other one.
- ❑ **Y resolution:** We can also set the Y resolution for our image here with the proper unit. We will leave this field as it is.
- ❑ **Unit:** This unit is for resolution. There is a wide variety of units available for this option. Also, there is a **More...** link. So, we can choose more units. We will leave this field as it is.
- ❑ **Color space:** We can choose RGB or Grayscale as our color space. We will select RGB.
- ❑ **Fill with:** This will be our fill up option. We will select **Transparency** to create transparent images. There is also an option for setting background colours with the **background** option.
- ❑ **Comment:** Here, we can write comments describing our image.

3. Click **OK**.

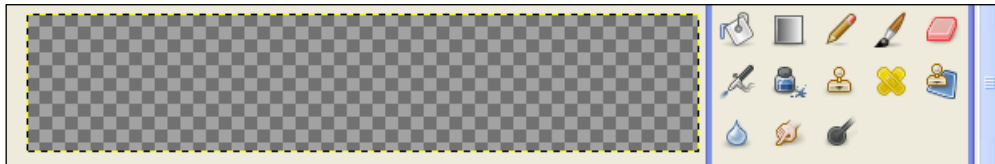
4. Our image area is small to view, so let's zoom in. For that we will go to **View | Zoom**. Here, you will see the following window:



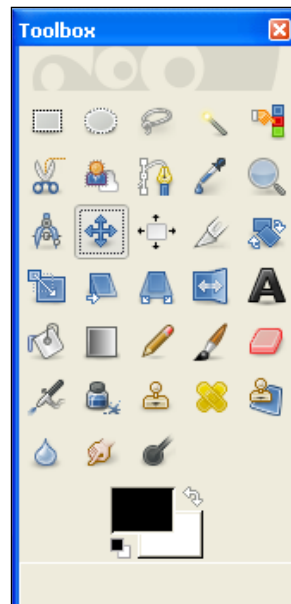
This shows options for:

- ❑ **Zoom in**
- ❑ **Zoom out**
- ❑ **Fit Image in Window**
- ❑ **Fill Window**
- ❑ Different ratios for zoom
- ❑ Custom ratios

5. We will set the zoom to **2:1 (200%)** ratio. Alternatively, we can click the **zoom** tool on the toolbox. Then, clicking on the image will zoom in and **Ctrl+click** will zoom out. So, our image area becomes like this:

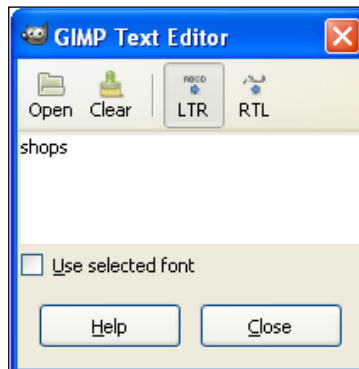


6. Then, we will write something for our logo. Click on the **A** or **Text Tool** on the **Toolbox**, or you can go to **Tools | Text**. At the bottom of the toolbox, we will see the control panel for texts. Let's discuss the options:



- **Font:** Here, we can select fonts from a wide range of values. Clicking on the button, will show the available options for text. We will choose **Lucida Calligraphy Italic**.
- **Size:** Select the appropriate size and size unit. The size will be 36px.
- There are hinting, force auto-hinting, and anti-aliasing options.
- **Color:** We will select **#1b931b** for our logo.
- **Justify:** There is left-justify, right-justify, centered, and filled options. We can also set indentation, letter-spacing, and line-spacing. We will choose left-justify, letter-spacing for 0.1px, line-spacing is 2px, indentation 1px.

7. Now, select an area, a window will open up for text input; write our project title **shops** as shown in the following screenshot:



8. We will select **Use selected text** on the window. Now, our logo is at this stage:



9. We will find an appropriate image of a shopping cart from the Web.
10. We will use it as a layer with our existing image. Go to **File | Open as Layers**. Then, choose the downloaded shopping cart image. Now, select open. Our image will be like this:

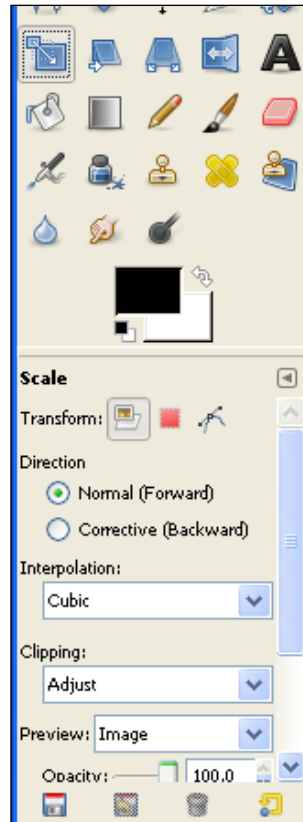


11. Now, we will select the 'move' tool from the toolbox.
12. Then, select the cart layer and move it to the right-hand side.



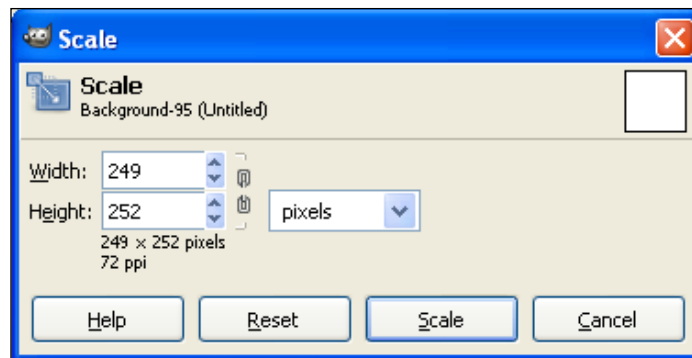
13. There is the scale tool, the perspective tool, the shear tool, and the flip tool. We are now going to discuss them one by one:

14. Scale tool: Using the scale tool, we can resize an image to proper size. We can see the control panel for scale at the bottom of the toolbox.



Let's discuss the options:

- ❑ **Transform:** This shows which options we want to transform.
 - ❑ **Direction:** Normal or corrective options are available.
 - ❑ **Interpolation:** There are none, linear, cubic, and sinc options for interpolation.
 - ❑ **Clipping:** We can set clip option to adjust, clip, crop to result, and crop with result.
 - ❑ **Preview:** We can preview as image, grid, image + grid, and outline.
 - ❑ **Opacity:** We will control opacity with this option.
15. To use scale, click and hold mouse on the image area that you want to scale and then drag the mouse in the direction to which you want to scale the image. We scale our image to a smaller one to adjust it with our logo. When we drag our mouse, there is another window displaying **x** and **y** values. We can manually set our values there too.



Our logo will be like this by this time:



16. Shear tool: We can change the magnitude of our image with the shear tool. By clicking the shear icon in the toolbox, we can see its control panel at the bottom of the toolbox. Shear tool has the same attributes as the scale tool. Now, our logo will look like this:



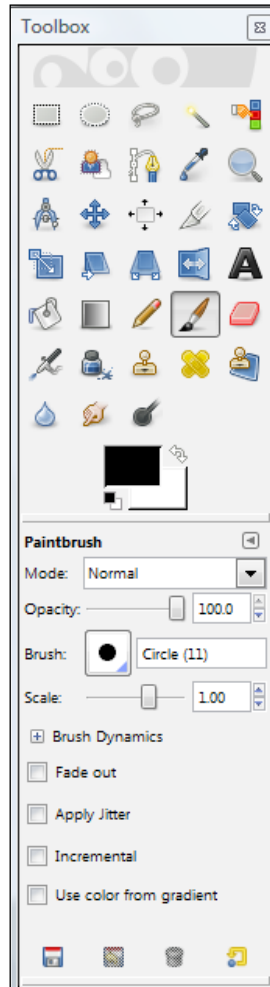
17. Perspective tool: This is an excellent tool for image modification. We can change our image at any corner. After selecting this tool, when you click on the image, you will see four boxes on the four corners of our image and a circle at the centre. We can drag any of those corners to any extent. There will be a window showing the transformation matrix. At this point in time, our logo is at this stage:



18. Now, we will use the 'smudge' tool. This will help us to smudge a portion selectable with a brush. Click the 'smudge' icon on the toolbox, then, we will move by clicking and holding the mouse on our image. We can also select this option with the key s or by going to **Tools | Paint Tools** options. So, we will get this kind of effect on our logo:



We see the control panel for the smudge option at the bottom of the toolbox. Let's discuss the options:



- ❑ **Mode:** There are different options for selecting mode. We will use **normal**.
- ❑ **Opacity:** This option sets the opacity of the image. We can set it manually or by using mouse on the roller. We will set the opacity to **20**.
- ❑ **Brush:** We have different kind of brushes available in GIMP. Click on the button beside the brush. It will open up a window showing all the brushes. We can change the view of brushes to grid, list, and also zoom in or zoom out. We see the name of each brush beside its image.
- ❑ **Scale:** This option sets the scale option of the smudge effect. We can set it manually or by using mouse on the roller. We will set the opacity to **0.17**.

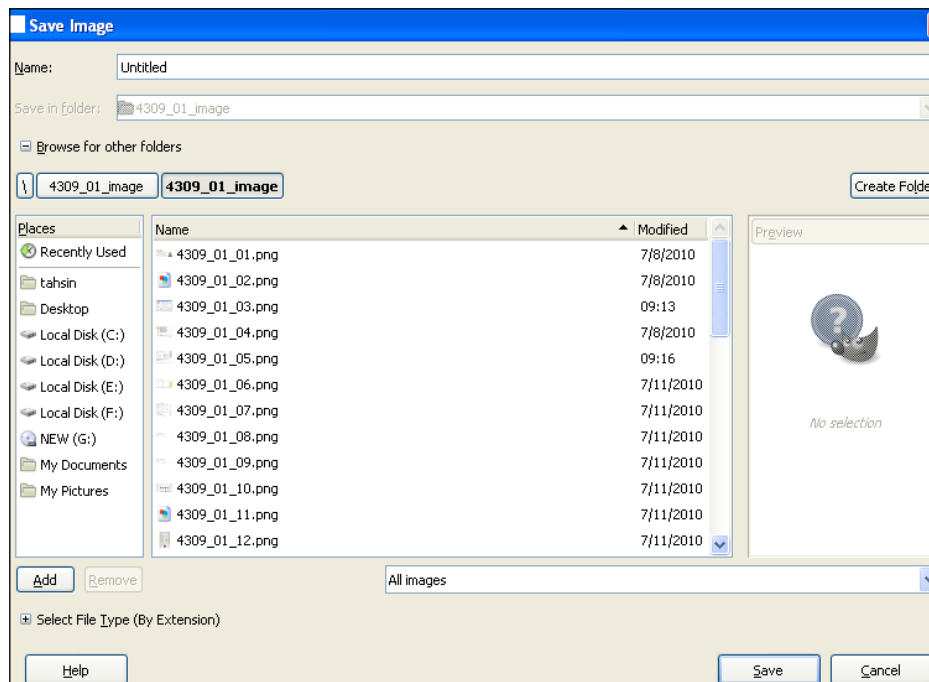
- ❑ **Brush dynamics:** We can set some dynamic options of brushes here. There are pressure, velocity, hardness, and rate.
- ❑ **Fade out:** We will set the length and unit for fade out selection.
- ❑ **Apply jitter:** This option sets the jitter option of the smudge effect. We can set it manually or by using the mouse scroller. We will set the opacity to **2.81**.
- ❑ **Hard edge:** We set the hardness of the edge option of the smudge effect. We can set it manually or by using mouse on the roller. We will set the opacity to **50.0**.

GIMP comes with different useful plugins. We need the 'export' plugin to save as a png file. We can get the plugin from <http://registry.gimp.org/node/33>. There is one for Windows and one for Linux. You can download the one that you need. We will download it for Windows, and extract the files. There are three files:

- ▶ Exe file: plug-in file.
- ▶ Readme file: it contains necessary information for installation.
- ▶ License file

We will copy and paste the exe file in to our GIMP install directory: E:\install\GIMP-2.0\lib\gimp\2.0\plug-ins. Then, restart GIMP.

Now, we will save our logo. Go to **File | Save as**, then the following window will open:



Let's discuss every field:

- ▶ **Name:** We will set the file name here. We can also set the extension optionally.
- ▶ **Save in folder:** The saving directory can be chosen here.
- ▶ **Browse for other folder:** We can choose our saving directory more clearly here. We can create a new folder too.
- ▶ **Select file type:** We will select the file extension type using this tool.

Then, we will click the **save** button.

There's more ...

There are other designing tools that will be helpful. They are:

- ▶ Photoshop
- ▶ Fireworks

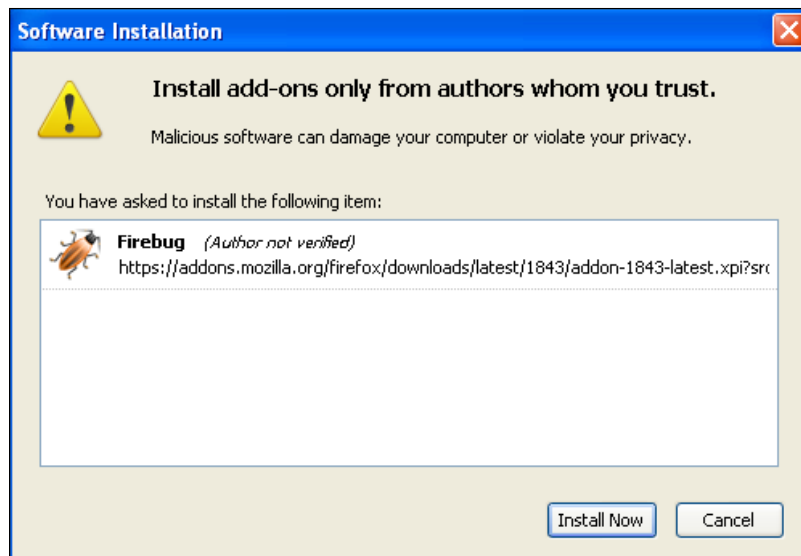
Definitely Photoshop is a very good tool for developers as it has many features in it. But we have gone for GIMP as it is open source. And also, GIMP is enriched with many features. There are many plugins for GIMP.

Another one is Fireworks. This is also a good option for designing.

Firebug

For our development, we need to install some Firefox add-ons. Firebug is one of them. Firebug is a very helpful tool for developers. We can inspect HTML and modify style and layout in real-time. It uses the most advanced JavaScript debugger available for any browser. With it, we can examine detailed measurements of our site's network activity. First, let's see how to install Firebug.

We need to go to this website to add the Firebug addon: <https://addons.mozilla.org/en-US/firefox/addon/1843/>. There is a button that says **Add to Firefox**. Current version works with Firefox 3.5 to 3.6.*. Click on the **Add to Firefox** button to add it. The following window will appear:



Wait until you see an **Install Now** button. Clicking the button will install Firebug, now we need to restart Firefox. Firefox now has a Firebug button at the bottom of the browser window.



Inspecting elements using Firebug

Get familiar with the Firebug tool by clicking the mouse pointer icon. Now, take the mouse on any HTML element on the browser. We will see the selected HTML element on the left side and corresponding style and layout on the right side.

For example, let's open our project on the browser. We will inspect its elements with Firebug.

How to do it...

1. Click the Firebug icon on the browser.
2. And then, click the **mouse pointer** icon of the Firebug console.
3. Move the mouse on the logo of our project.

We will see the corresponding HTML element on the left side of Firebug.

4. On the right side, we can see the corresponding style of the element by clicking the style tab. We can edit the style values there and also examine the new one. We must remember that these editions are not permanent, so, we need to write the appropriate editions into our code.
5. Let's do some adjustment of our logo. Change the values of height and width of CSS style on the right side.
6. On hovering over any CSS style, you will see an **off** icon beside each style. If we select it, the selected style will be turned off. We can make it available by selecting it again.
7. We can change the padding and margin by selecting the layout tab.
8. On the left side, we will select the logo image. Then, click on the **edit** button. Here, we can change the HTML properties also.

There's more...

We have shown only Firefox add-ons in this chapter. But there is Firebug lite, which can be used with many other versions.

Installing Web Developer

Another important add-on for Mozilla is Web Developer. We can edit CSS files using this tool and we can also inspect HTML elements with it. Web developer has a collection of features for developers.

Getting ready

Let's install web developer first. We can add it from this website link:
<https://addons.mozilla.org/en-US/firefox/addon/60/>.

How to do it...

There is an **Add to firefox** button. Click on the button. It will prompt with an **Install Now** button.

Click the button. Then the installation will start. To get Web developer in action, we need to restart our browser.

We are going to inspect our project 'shop' with Web developer. First, open our site in Firefox. Now, we will use Web developer.

How it works...

There are different types of tabs in the Web developer toolbar:

1. First, we will click on the **Image** tab. This has different types of properties for image inspection. Now, if we select **Display Image Dimension**, it will show the size of images.
2. We can see the file size of each image with the **Display Image File Sizes**.
3. We can also examine different attributes of the image tag. To see the image path, select **Display Image Path**.
4. We can see all the image-related information in a page by clicking **View Image Information**. This will show us the image path, height, width, number of images on the page, file size, and the alt attribute.
5. We have other options to examine the image information on our site, like **Hide Image**, **Hide Background Images**, and so on.
6. Now, we will inspect **Information**. There are many options in this section:
 - ❑ First, is the **Display block size**. It outlines all our blocks.
 - ❑ The next necessary option is **Display Color Information** under the **Information** tab. It will open a new tab and show the colours used in this website. So, using this tool, we can understand what colours are used in a website.
 - ❑ The **View Javascript** option will help us examine what JavaScript a site has and also, it is useful to debug a JavaScript error.
 - ❑ The **View Page Information** will show page-related information. If we want to have some images from other sites, then this is a good tool to use. After clicking **View Page Information**, click on **Media**. There, we will see all the images of that page. Select the one you want and click **Save as**.
7. Under the **Miscellaneous** section, we have the **Edit Html** option. We can edit our HTML, on the browser, with it. Let's do something in our project with it. We will wrap the titles with `<h1>` tag, and see what it shows in the browser. Using this tool, we can easily examine a site's structure and fix it on the fly.
8. In the **Outline** section, we have the **Outline Block Level Elements**. This gives us the layout structure of a website.
9. Now comes a very important one, the **CSS** tab. We can edit CSS using this option. Click **Edit css**, this will open up a new section in the browser containing all our CSS for the page. We can select the CSS that we want to edit. When we edit, we can see the result of our editing instantly.
10. The **View Style Information** option shows us in detail the style information about an element. If we select the header, it will show the following information. So, when a website catches our eye, we can easily examine their style information with it.

Installing Colorzilla

Colorzilla is another important Firefox addon for site development. With Colorzilla, we can get a color reading from any point in our browser, quickly adjust this color, and paste it into another program. At times, we may not know the code of the colour. Here Colorzilla comes into play. Let's first see its installation process.

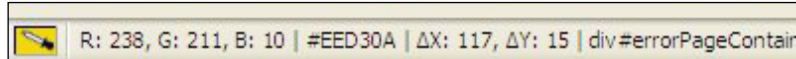
How to do it...

To download Colorzilla, we need to go to this link: <https://addons.mozilla.org/en-US/firefox/addon/271/>. We will add this addon with the same process as the previous two addons.

Click the **Add to Firefox** button. Follow the process mentioned previously to install Firefox addons. We must remember that we need to restart our browser to use Colorzilla's features.

We are going to inspect our project 'shop' with Web developer. First, open our site in Firefox. Now, we will use Web developer.

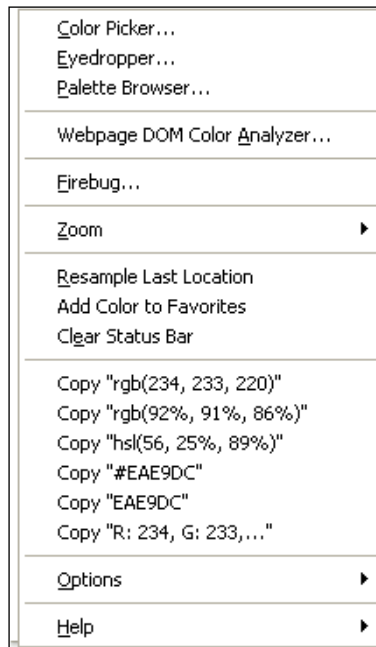
On the left hand side of our status bar, there is a Colorzilla icon. By clicking on the **Colorzilla** icon, if we select anything on the web page, then the status bar will show information of the colour code of this element. It will look like the following:



So, what we are seeing here is:

- ▶ **Colour:** The first one is showing the colour itself.
- ▶ **RGB value:** It shows the RGB value of our selected element colour.
- ▶ **Colour code:** Next, it shows the corresponding colour code.
- ▶ **Point:** It shows the x, y coordinates of our selected element.
- ▶ **Div:** The selected element's div.

By right-clicking on the Colorzilla icon, we can see more options for Colorzilla. Let's discuss them:



- ▶ **Color picker:** It will open up a colour picker window.
- ▶ **Eyedropper:** This enables us to pick the colour of an element.
- ▶ **Webpage dom color analyzer:** The colours that are used in this web page.
- ▶ We can add colours to favourites with it.

There's more...

IE tester

A good number of users use Internet Explorer. Some features of CSS and JavaScript may not work properly in different versions of IE. So, we need to test our website in some versions of IE. IETester enables us to test our site with different versions of IE. It supports version 5.5 to IE 9 preview. We can download it from here: <http://www.my-debugbar.com/wiki/IETester/HomePage>.

Latest Designing trends

When we develop a site, we must take into consideration the latest designing trends. The designing trends change from time to time. So, in order to make our site successful, we need to know the designing trends of the current time. Here, we will discuss some key areas to look at when developing a site. A site is made up of three common sections:

- ▶ **Header:** Nowadays, stretched headers are becoming popular. So, in this book, we will also see how to make them. Stay with us! Some headers have a navigational block and authentication links. We are going to follow them. Users love a navigational block in the header area.
- ▶ **Content:** This section holds body contents. According to the latest designing trend, it is a good idea to show people the main logic in the home page without scrolling down the page. Look at this site:



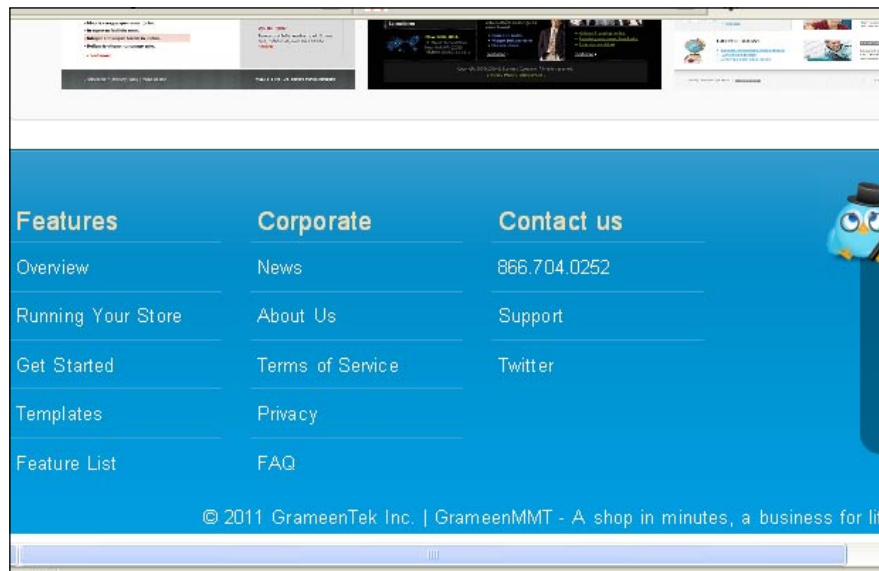
We are getting the basic information without scrolling down the homepage. Another concern will be to show slider, carousels, videos, and so on in the home page. These give a good impression of the site to users. There is also the following layout:

- ❑ **Three column:** There will be three columns in our layout. Normally, news websites have a three column layout to show more information.
- ❑ **Two column:** Maximum websites have a two column structure. It has a left column and a right column layout.
- ❑ **Single column:** Usually, it is a good idea to show a single column home page with the main featured images. We can use sliding for the images to appear on the home page.



The above image shows show a four column layout of home page. We can again show a combination of a four column block, two column block, or any other combination on a page.

- Footer: This is the bottom part of the site. Wide and long footers are gaining popularity day by day. Following is a popular footer:



We can use twitter tweets, news block, social networking blocks, and the site's internal links on the footer.

2

Store Decoration

In this chapter, we will cover:

- ▶ Changing your shop's general settings
- ▶ Modifying the settings of your store
- ▶ Basic localization settings
- ▶ Setting image properties
- ▶ Changing the administrator favorites icon
- ▶ Setting up the SMTP mail server
- ▶ Modifying server options
- ▶ Creating categories
- ▶ Adding products to a category

Introduction

In the previous chapter, we set up our environment. Our store 'shop' was created. Now, we will decorate our store. In this chapter, we will see how to change store logos, upload products, create categories, and so on.

We will see here how we can customize store settings with the admin panel. We will upload our logo and a favicon for our store and admin panel.

Entering into the administrator panel

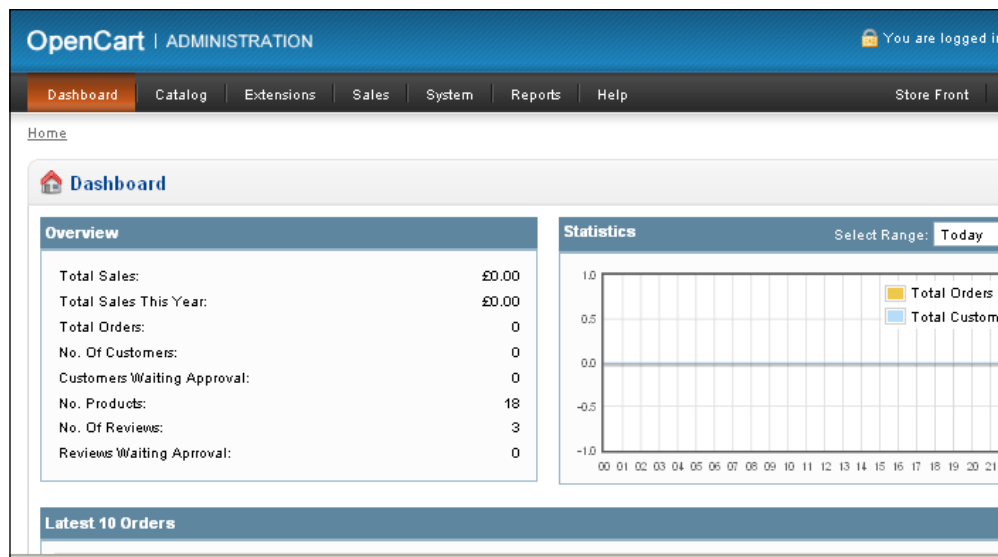
For our store modification, we need to enter into the administration first. OpenCart store frontend and administration are two different sections. We will find the `admin` folder under the `shop` folder. It is a good practice to rename our `admin` folder. This will minimize security leaks.

We will rename it `shop_back`. And then, we will also need to change the `config.php` file under the `admin` folder. Rename the `admin` folder to `shop_back`. To enter into the admin panel, we need to add `shop_back` at the end of our store URL. So, our admin URL will be `http://local.shop.com/shop_back`. We will be greeted with the following window:



The image shows the OpenCart Administration login page. It has a blue header with "OpenCart | ADMINISTRATION". The main content area is white and contains a login form titled "Please enter your login details." with a padlock icon. The form has fields for "Username:" and "Password:", and a "Login" button. At the bottom, it says "OpenCart © 2009 All Rights Reserved."

We will enter our admin username and password. Then, we will see the following admin page. This is the home page of the admin panel:



Changing your shop's general settings

Our first recipe for this chapter will be modifications of our shop's general settings through the admin panel. Here, we will set the basic settings for our store. We will set the store's name and address. Also, we will set the contact e-mail address in this section. We can modify this information at a later stage too.

Getting started

We just created our OpenCart shop. But, we need to change its default settings. So, we will go to our admin panel, and enter the username and password.

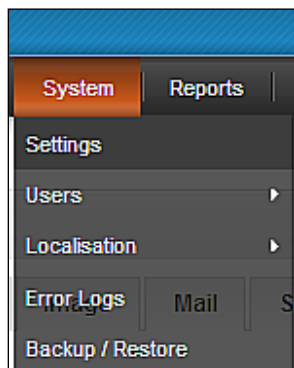
We will store our admin panel username and password for future use:

Username: _____.

Password: _____.

How to do it

You can see the top admin navigation on the admin panel. First, we need to go to **System | Settings**.



Store Decoration

Then, we will see a form containing the general information of our store 'shop'. We will modify that information according to our need. First, we will customize the **General** settings. This window will have the general settings for our store. The window will be like the following:

The screenshot shows a 'Settings' window with a title bar containing a gear icon, the text 'Settings', and buttons for 'Edit Store: Default', 'Create A New Store', 'Save', and 'Cancel'. Below the title bar is a tabbed interface with tabs for 'General', 'Store', 'Local', 'Option', 'Image', 'Mail', and 'Server'. The 'General' tab is active. It contains several form fields with red asterisks indicating required fields:

- * Store Name:** A text box containing 'Your Store'.
- * Store URL:** A text box containing 'http://localhost/openc.'. Below it is a small text note: 'Include the full URL to your store. Make sure to add "/" at the end. Example: http://www.yourdomain.com/path/'.
- * Store Owner:** A text box containing 'Your Name'.
- * Address:** A large text box containing 'Address 1'.
- * E-Mail:** A text box containing 'tdd@ddd.com'.
- * Telephone:** A text box containing '123456789'.

We can see our changes in the **CONTACT US** page. Click the **Contact** button on top of it. We will see this window:

The screenshot shows a web page layout with three main sections: 'CATEGORIES', 'CONTACT US', and 'SHOPPING CART'. The 'CONTACT US' section is the central focus and contains the following elements:

- Address:** A text box containing 'Your Store' and 'Address 1'.
- Telephone:** A text box containing '123456789'.
- First Name:** A text box.
- E-Mail Address:** A text box.
- Enquiry:** A large text box.
- Footer:** A text box with the prompt 'Enter the code in the box below:'.

The left sidebar contains 'CATEGORIES' (Desktops, Laptops & Notebooks, Components, Software, Phones & PDAs, Cameras, MP3 Players), 'BRANDS' (a dropdown menu with 'Please Select'), and 'INFORMATION' (About Us, Privacy Policy, Terms & Conditions, Contact Us, Site Map). The right sidebar contains 'SHOPPING CART' (0 items), 'BESTSELLERS', and 'FEATURED' (Apple Cinema £100.00, HTC Touch H £569.22, MacBook £587.50).

How it works...

Now, let's discuss every field:

- ▶ **Store Name:** The name of our store will be set here. We will set to 'shop'. We can change it any time.



- ▶ **Store URL:** We will write our store URL here. This will be `http://local.shop.com/`. Note that we need to add a trailing '/' in the url.
- ▶ **Store Owner:** The store owner's name will be set here. This can be your name or you can also set it as 'shop' and so on.
- ▶ **Address:** We will store the location of our store here. This will be the physical address of the store.
- ▶ **E-Mail:** The admin email address will be set here. We will set it as `admin@shop.com`. Our customers will send emails to this email ID.
- ▶ **Telephone:** The telephone number of store admin will be set in this field. We will add the international country code also.
- ▶ **Fax:** The Fax number of store admin will be set in this field. For now, we will set a dummy value.

Now that we have set all our information, we will save it. Click the **Save** button on top of the form.

A screenshot of a form's header section. It contains the text 'Edit Store:' followed by a dropdown menu showing 'Default'. To the right of the dropdown are three buttons: 'Create A New Store' (blue), 'Save' (blue), and 'Cancel' (blue).

Modifying the settings of your store

We have set the general information; now, we will set store-specific information. Let's see this recipe now.

Getting started

To change store settings, we need to log in to our admin panel. We will use our previously saved admin username and password for login.

How to do it...

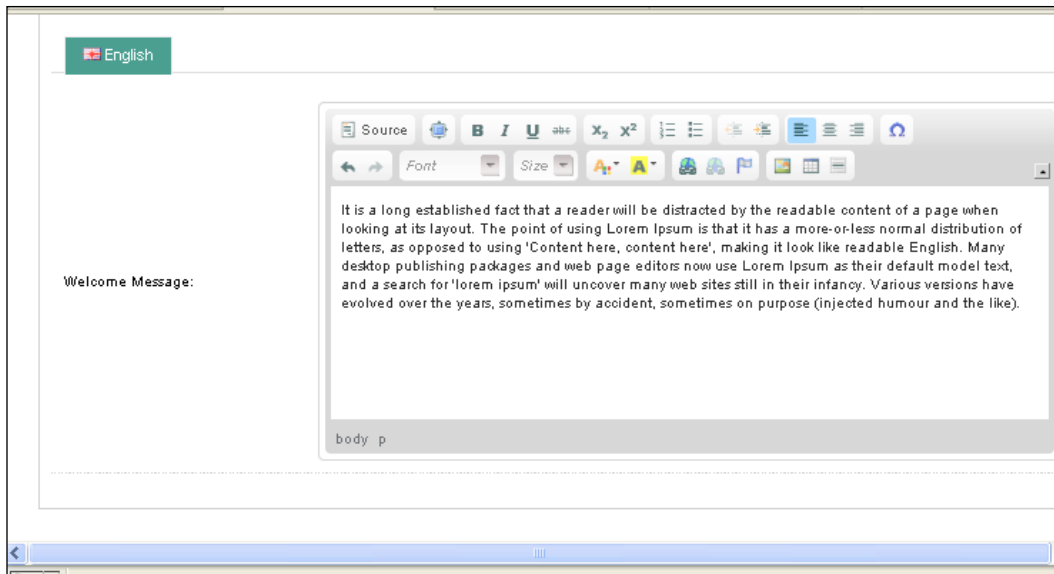
We will click the **Store** tab under **System | Settings** section. The following window will open:

The screenshot shows a settings window with the following elements:

- Tabs: General, **Store**, Local, Option, Image, Mail, Server
- * Title: Your Store
- Meta Tag Description: My Store
- Template: shop (with a dropdown arrow)
- Image upload icon (camera)
- Bottom bar: Save icon

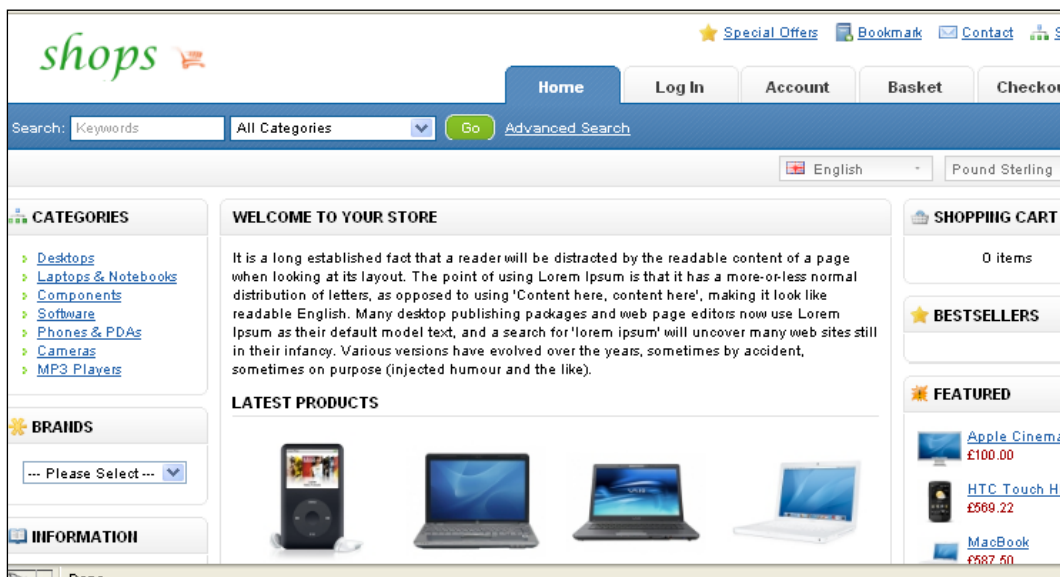
Let's discuss them:

- ▶ **Title:** Our store's title will be set here. This title will be set for our home page. We will set it as 'shop'.
- ▶ **Meta Tag Description:** For search engine optimization, we will set appropriate Meta tag description in this field.
- ▶ **Template:** We can change the template of our site through this drop-down box. Currently, it will show our default template. When we add new templates, they will be shown here.
- ▶ **Welcome Message:** We will write a welcome message for our customers using this text area. For example, we will write a dummy text from 'lorem ipsum'. To take some dummy text, go to the URL <http://www.lipsum.com/>. We have our dummy text, that we will now paste in the text area. This message will appear on the home page of our store. We have different controls for this field. So, we can format our text as and when necessary using those tools.



We have different controls for this field. So, we can format our text as we like using those tools.

Finally, we will save it as we have done before using the **Save** button on the top right side. Now, we will go to the store front to view the changes:



Here, in the above picture, we can see our dummy welcome messages.

Basic localization settings

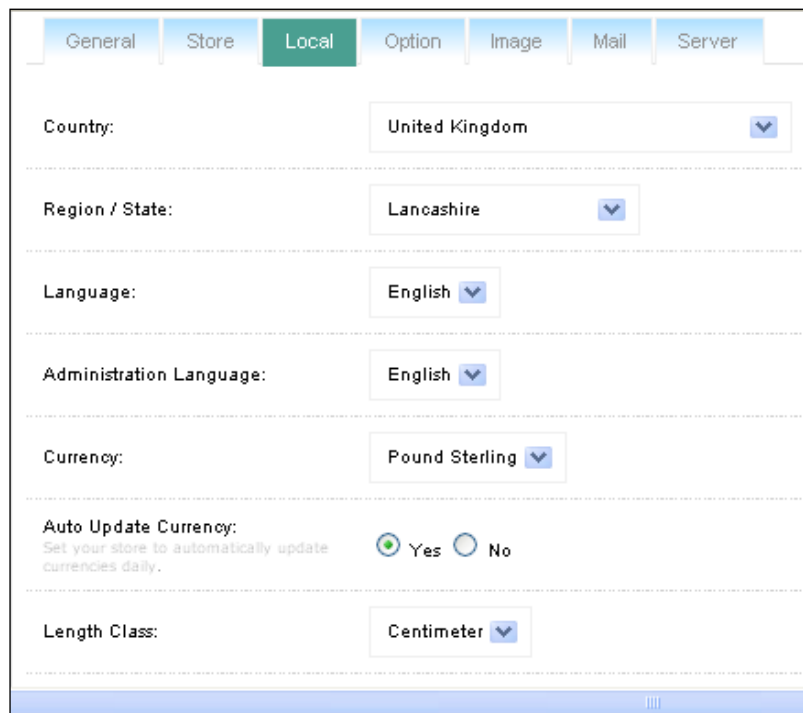
We will set our basic localization settings in this recipe. We will see a detailed level of localization later in this book. So, stay with us for that. But for now, we will stick with the basic localization settings only.

Getting started

We will go to our admin section with our stored admin username and password. Then, we will go to **System | Settings**, and under it, we will select **Local** tab.

How to do it

Choosing the **local** tab, we will see the basic localization window before us.



The screenshot shows the 'Local' settings tab in an admin interface. The tabs at the top are General, Store, Local (selected), Option, Image, Mail, and Server. The settings are as follows:

- Country: United Kingdom (dropdown)
- Region / State: Lancashire (dropdown)
- Language: English (dropdown)
- Administration Language: English (dropdown)
- Currency: Pound Sterling (dropdown)
- Auto Update Currency: Set your store to automatically update currencies daily. (Yes/No radio buttons, Yes is selected)
- Length Class: Centimeter (dropdown)

We will see each field here:

- ▶ **Country:** We will set the country in which our store is located. There is a list of all the available countries.
- ▶ **Region/state:** After choosing the country, this field will be populated automatically. Then, we can select the appropriate one.
- ▶ **Language:** We will choose the appropriate language for our store. In this case, we have only **English** available to us.
- ▶ **Administration Language:** We will also choose the appropriate language for our store administration section. In this case also, we have only **English** available to us.
- ▶ **Currency:** By default, we have three currencies available: euro, pound sterling, and dollar.
- ▶ **Auto Update Currencies:** Currency rate changes frequently. OpenCart comes with a great feature that it will automatically update the currency values with the market price. If we select **Yes**, then it will work automatically.
- ▶ **Length class:** We will choose the length measurement unit in this field. We have three values available currently: centimetre, millimetre, and inch. We will leave it as it is—or you can choose one from the list.
- ▶ **Weight class:** We will also choose the weight measurement unit in this field. We have four values available currently: kilogram, gram, pound, and ounce. We will leave it as it is.

After setting that information, we will save it using the **Save** button. We can see the store front to view the changes.

There's more

We can customize different options for our shop. In this section, we will set the tax display option, invoice initial number, selecting customer group, guest checkout option, and so on. We can also see the agreement on the terms and privacy policy of our store. This section enables product download option and also sets the download order status.

Here, we will set how many items we want displayed in the frontend and also in the admin section.

Setting up other options

For this recipe, we need to log in as admin using our stored credentials.

We need to go to the **System | Settings** section and then select the **Option** tab. The following window will open:

General	Store	Local	Option	Image	Mail	Server
<p>* Default Items per Page (Admin): Determines how many admin items are shown per page (orders, customers, etc)</p> <input type="text" value="20"/>						
<p>* Default Items per Page (Catalog): Determines how many catalog items are shown per page (products, categories, etc)</p> <input type="text" value="12"/>						
<p>Display Prices With Tax: <input checked="" type="radio"/> Yes <input type="radio"/> No</p>						
<p>Invoice Start No.: Set the starting number the invoices will begin from.</p> <input type="text" value="001"/>						
<p>Invoice Prefix: Set the invoice prefix e.g. IN/001</p> <input type="text" value="IN/"/>						
<p>Customer Group: Default customer group.</p> <input type="text" value="Default"/>						
<p>Login Display Prices: Only show prices when a customer is logged in.</p> <input type="radio"/> Yes <input checked="" type="radio"/> No						
<p>Approve New Customers: Don't allow new customer to login until their account has been approved.</p> <input type="radio"/> Yes <input checked="" type="radio"/> No						
<p>Guest Checkout</p>						

Lets' see the different fields:

- ▶ **Display Prices with Tax:** There is a yes/no radio button to select the option. Selecting **Yes** will show the product price including the product tax value. We will select no so that it shows less price initially.
- ▶ **Invoice Start No:** For our order invoices, we will set a starting number. We will set it to 007.
- ▶ **Invoice Prefix:** This will show the invoice prefix of our site. We will set our invoice prefix as **INP**.
- ▶ **Customer Group:** We can select a customer group here. Currently, there is **Wholesale** and **Default**. We can add other customer groups. We will see that later in this book. But, for our store, we will set the default value.
- ▶ **Login Display Prices:** This option asks whether we need authentication to show the prices of a product. We will choose **Yes**, because this will increase our users on the store.

- **Approve New Customers:** This option will not allow a new customer to log in until their account has been approved. Choosing **Yes** may delay the registration procedure. It may discourage a user to register. So, we will set it to **no**, and then users can log in after creating their account.
- **Guest Checkout:** We will allow customers to check out without creating an account. This will not be available when a downloadable product is in the shopping cart. This option will give our customers a hassle free shopping experience. We will set it to **Yes**. When customers check out, they will see this page:

I am a new customer.

Checkout Options:

☒ Register Account

☐ Guest Checkout

By creating an account you will be able to shop faster, be up to date on an order's status, and keep track of the orders you have previously made.

Continue

Then, we need to fill up our information with the provided form, like this:

CREATE ACCOUNT

If you already have an account with us, please login at the [login page](#).

Your Personal Details

* First Name:

* Last Name:

* E-Mail:

* Telephone:

Fax:

Your Address

Company:

* Address 1:

Address 2:

* City:

Post Code:

- **Account Terms:** When a customer creates his/her account, he/she needs to agree with our policies. In this option, we will set which policies they need to abide by. It may be **Terms & Conditions**, **Privacy policy**, or **About Us**. We will set it as the default. So, this will force customers to agree with our privacy policy while they create their account with our store **shop**.

- ▶ **Checkout Terms:** Similar to the previous option, this field sets which agreement our customers need to abide by before checkout. It has the same three options as the previous one.
- ▶ **Display Stock:** We are not going to display stock quantity in the product page. So, we will select **No**. But, you can choose other options if you want. And this is also true for the other fields of the form. We are just seeing how to make a simple store here.
- ▶ **Show Out Of Stock:** This sets the message for when the product is out of stock. We will set it to **Yes**.
- ▶ **Stock Checkout:** This will allow customers to still check out even if the products they are ordering are not in stock. We will set it to **No**.
- ▶ **Stock Subtract:** We will set this to **Yes**, so, when an order is processed, stock quantity will be subtracted.
- ▶ **Order Status:** This will set the default order status when an order is processed. We will leave it as it is.
- ▶ **Allow Downloads:** OpenCart comes with a product download feature. This option enables us to download a product. We will set it to **Yes**.
- ▶ **Download Order Status:** We will set the **Complete** status, when a download will be complete.
- ▶ **Default Items per Page (Admin):** This attribute sets the number of items that will be shown on the admin page. We will leave it as it is.
- ▶ **Default Items per Page (Catalog):** This attribute sets the number of items that will be shown on the catalog page. We will leave it as it is.
- ▶ **Display Weight on Cart Page:** This sets whether we will show the weight value in the cart page. We will set it to **Yes**.
- ▶ **Use Shipping Session:** This enables us to automatically populate shipping values with session. We will set it to **Yes**.

We are done with setting different options for our shop. Now, we will save it. After successful submission, we will see this message:

Success: You have successfully saved your settings!

See also

- ▶ Changing your shop's general settings

Setting image properties

So far, we have modified lots of store properties. Now, we will set the image properties for our site. We can upload store logos and also, we can change the favicon for our store. We have already created our logo. Now, we will upload it and set it as our store logo. Here, we also change the dimension of different images.

Getting started

For this recipe, we need to log in as admin using our stored credentials. We have already shown the admin login process.

How to do it

1. We need to go to the **System | Settings** section and then select the **Image** tab. This will show the following window:

The screenshot shows the 'Image' settings tab. At the top, there are tabs: General, Store, Local, Option, Image (selected), Mail, and Server. Below the tabs, the 'Store Logo' section shows a preview of a logo with the text 'shops' and a shopping cart icon. Below that, the 'Icon' section shows a preview of a shopping cart icon and a note: 'The icon should be a PNG that is 16px x 16px.' Below the icon section, there are five rows of settings, each with a label, a red asterisk, and two input fields for width and height:

- * Product Image Thumb Size: 250 x 250
- * Product Image Popup Size: 500 x 500
- * Category List Size: 120 x 120
- * Product List Size: 120 x 120
- * Additional Product Image Size: 150 x 150

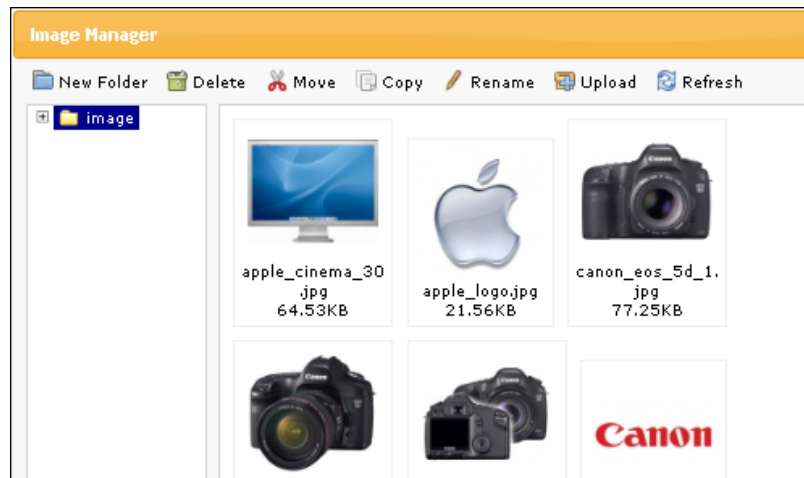
At the bottom right of the form, there is a blue button with a white icon.

We will now discuss the attributes:

2. **Store Logo:** We will replace the store logo using the **Upload** option. Let's replace it with our logo that we created in the previous chapter. Click on the upload icon.



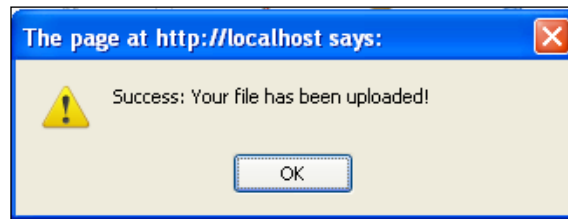
Then, it will open up the following window:



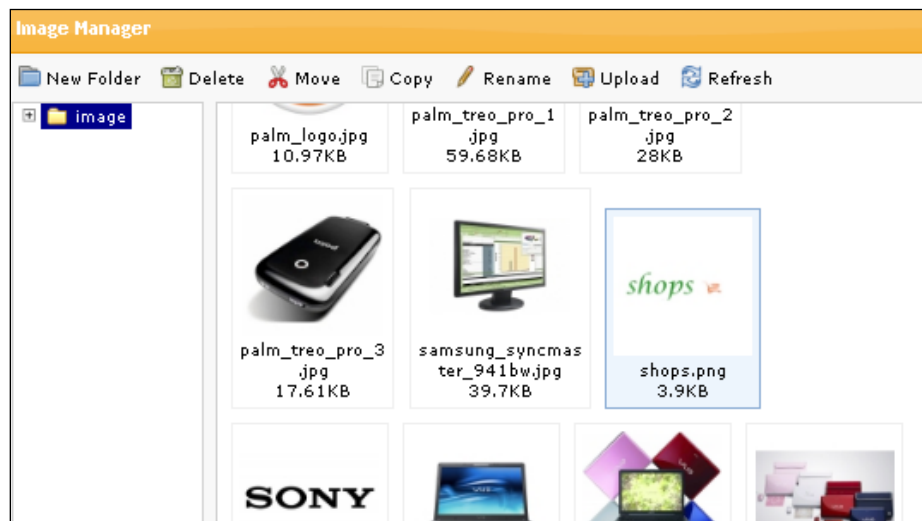
3. There is an **Upload** button on top of the window. This will lead to the following window:



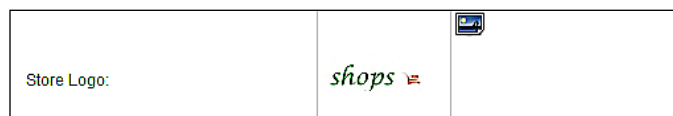
- We will choose our image, select it, and then click the **Open** button. If our image is uploaded successfully, we will be greeted with the following message:



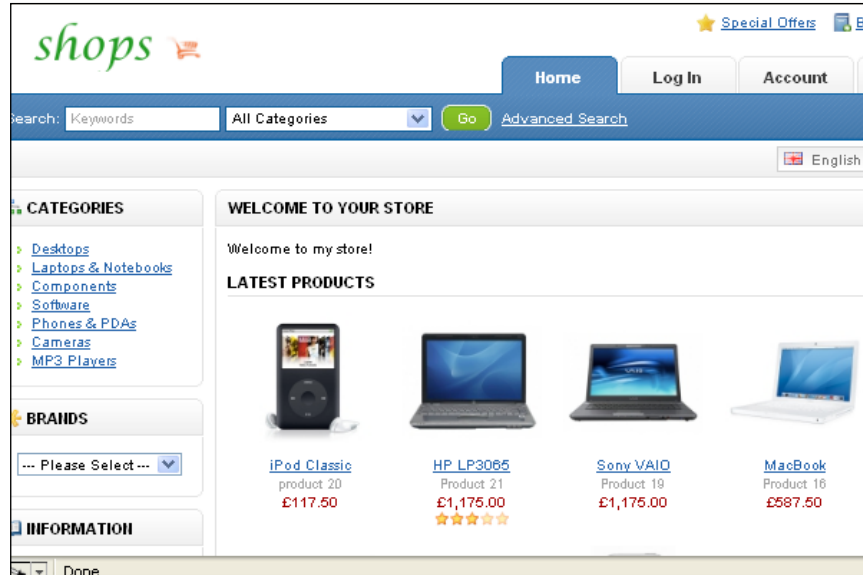
- Click **Ok**. Now, click **Refresh** on the top of the window. We will see our logo on the window, like this:



- We will select our logo and double click it. And, it will be set as our store logo.



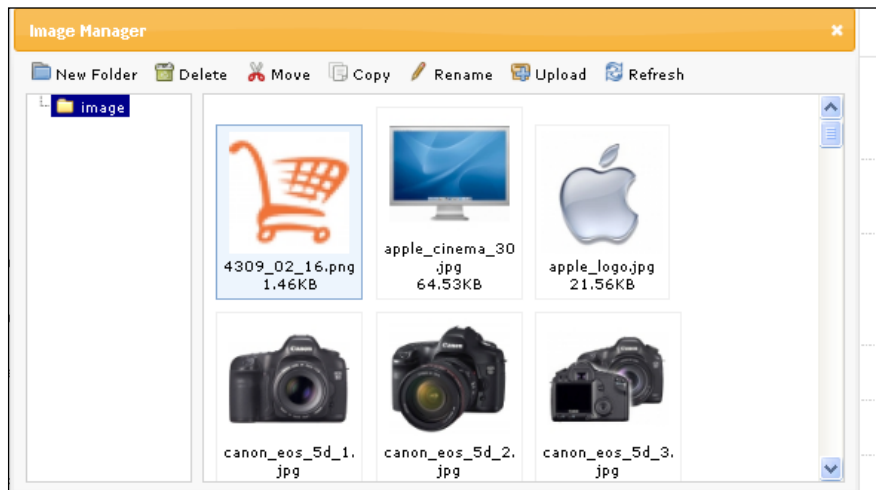
7. We need to save the changes to view our logo in action. Now, we will go to our store front by clicking on the **store front** tab on top of the admin panel.



8. Icon: We will upload the favicon image with this option. We will change our store favicon in the same way as we did with our store logo. We will use the following image as our store favicon:



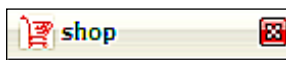
After uploading, we will see it like this:



9. Now double click it and then save it. So, our favicon is set and we have this view:



Also, when we go to our store front, we will see the changed favicon on our site, like this:



- ▶ **Product Image Thumb Size:** We will set the product image thumb size. Here, we can set the height and width of the thumbnail image. This image will be shown on the product detail page. We will leave it as it is.
- ▶ **Product Image Popup Size:** This will be the popup box image size. It is good to set it to a large size. For now, we will set it as default.
- ▶ **Category List Size:** We will define the category list size here. We will leave the dimension as it is. This image is for categories.
- ▶ **Product List Size:** This will set the product list dimension. This size will be applicable for the list view of products. The 'list view' refers to showing a list of products on the basis of some common properties like 'category'.
- ▶ **Additional Product Image Size:** Here, we can set this dimension for additional images of a product.
- ▶ **Related Product Image Size:** We will set the image sizes for the related products under the product detail page.
- ▶ **Cart Image Size:** This will set the product image size in the cart page. We will leave it as it is.

Changing administrator favorites icon

We have changed our store front favicon. OpenCart has an user interface for changing store front favicon. But, it does not have an interface in the admin panel for changing the admin favicon.

We will change our admin favicon by modifying our code. We will use the following image as our favicon:



First, we will go to the `admin/view/image` and paste this image in this folder. Now, we will make a change in the `header.tpl` file in the admin section. For that we need to go to `admin/view/template/common`. And add this line after the `<base>` tag in the header:

```
<link href="view/image/shop.png" rel="shortcut icon" type="image/x-
icon" />
```

Here, the `href` attribute contains the location of the image. For favicon, we use `shortcut icon` `rel` attribute. So, our `head` tag now becomes like this:

```
<head>
<title><?php echo $title; ?></title>
<base href="<?php echo $base; ?>" />
<link href="view/image/shop.png" rel="shortcut icon" type="image/x-
icon" />
<?php foreach ($links as $link) { ?>
<link href="<?php echo $link['href']; ?>" rel="<?php echo
$link['rel']; ?>" />
<?php } ?>
<link rel="stylesheet" type="text/css" href="view/stylesheet/
stylesheet.css" />
<link rel="stylesheet" type="text/css" href="view/javascript/jquery/
ui/themes/ui-lightness/ui.all.css" />
<?php foreach ($styles as $style) { ?>
<link rel="stylesheet" type="text/css" href="view/stylesheet/<?php
echo $style; ?>" />
<?php } ?>
<script type="text/javascript" src="view/javascript/jquery/jquery-
1.3.2.min.js"></script>
<script type="text/javascript" src="view/javascript/jquery/ui/ui.core.
js"></script>
<script type="text/javascript" src="view/javascript/jquery/superfish/
js/superfish.js"></script>
<script type="text/javascript" src="view/javascript/jquery/tab.js"></
script>
<?php foreach ($scripts as $script) { ?>
<script type="text/javascript" src="view/javascript/<?php echo
$script; ?>"></script>
<?php } ?>
</head>
```

Now, we will go to the browser and reload our admin section. Then, we will see the new favicon in the admin panel.



How it works . . .

Here, we have added a `link` HTML tag with the attribute `rel`. We set `shortcut icon` to define it as a favicon. Then, we set the appropriate image path.

Setting up SMTP mail server

We will have to send emails to the customers of our store. So, we need to configure our mail server. For that purpose, we need to set an SMTP (Simple Mail Transfer Protocol) server.

Getting started

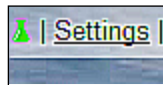
To set our SMTP information, we will create an email account for our store with Gmail.

How to do it

We will go to `http://gmail.com`. Then, we will create an email account with them.

We will fill up the form with the necessary information. And then click on the **Create my account** button.

Now, we will enter our newly created Gmail account and do some adjustments to our settings. Click **Settings** on the top-right corner.



Then, we will select **Forwarding and Pop/Imap**, and we will enable POP and IMAP settings.

Tip: You can also forward only some of your mail by creating a filter!	
POP Download: Learn more	1. Status: POP is enabled for all mail that has arrived since 5/14/08 <input type="radio"/> Enable POP for all mail (even mail that's already been downloaded) <input type="radio"/> Enable POP for mail that arrives from now on <input type="radio"/> Disable POP
	2. When messages are accessed with POP <input type="text" value="keep Gmail's copy in the Inbox"/>
	3. Configure your email client (e.g. Outlook, Eudora, Netscape Mail) Configuration instructions
IMAP Access: <small>(access Gmail from other clients using IMAP)</small> Learn more	1. Status: IMAP is disabled <input type="radio"/> Enable IMAP <input checked="" type="radio"/> Disable IMAP
	2. Configure your email client (e.g. Outlook, Thunderbird, iPhone) Configuration instructions
<input type="button" value="Save Changes"/> <input type="button" value="Cancel"/>	

After doing our changes, we will save it by clicking the **Save** button.

There's more...

We have created our account with Gmail. Now, we will set our mail options for our shop with the SMTP options that we just created.

Setting mail options

To set mail options, we need to log in as admin with our store. Enter the admin username and password.

We will go to **System | Settings**, and then, we will click on the **Mail** tab. It will open up the following window:

The screenshot shows the 'Settings' window with the 'Mail' tab selected. The 'Edit Store' dropdown is set to 'Default'. There are buttons for 'Create A New Store', 'Save', and 'Cancel'. The 'Mail' tab is active, showing the following fields:

- Mail Protocol:** A dropdown menu set to 'Mail'. A note below it says: 'Only choose "Mail" unless your host has disabled the php mail function.'
- SMTP Host:** An empty text input field.
- SMTP Username:** An empty text input field.
- SMTP Password:** An empty text input field.
- SMTP Port:** A text input field containing the value '25'.
- SMTP Timeout:** A text input field containing the value '5'.
- Alert Mail:** A section with the text 'Send a email to the store owner when a new order is created.' and two radio buttons: 'Yes' (selected) and 'No'.

Now, let's discuss the fields in it;

- ▶ **Mail Protocol:** We will choose the **Smtp** option from the drop-down box. The other available option is **Mail**.
- ▶ **SMTP Host:** This will be our SMTP host address. As we are using Gmail as smtp server, our host address will be `ssl://smtp.googlemail.com`.
- ▶ **SMTP Username:** This will be our Gmail username, for example, we set it as `shops@gmail.com`.
- ▶ **SMTP Password:** This is the Gmail password for our account.
- ▶ **SMTP Port:** We will set the SMTP port for the Gmail server. This will be 465.

- ▶ **SMTP Timeout:** We will leave it as it is.
- ▶ **Alert Mail:** Setting this option to **yes** will send us email when a new order has been processed.

After setting the fields, we will save it by clicking the **Save** button. And, we are done with the mail options.

Modifying server options

Now, we will modify some server settings. When we go into production, we can log errors on a log file. Here, we can also set the output compression level.

For setting a store, SSL-enabled service is pretty important. This will make our store secure. The transactions will be secure. We can enable SSL service for our store in this section.

Getting started

To set server options, we need to log in as admin with our store. We will use our stored credentials for login.

How to do it

We will go to **System | Settings**, and then click on the **Server** tab. It will open up the following window:

The screenshot shows the 'Settings' page in a Magento 2 admin interface. The 'Server' tab is selected. The page includes a top navigation bar with 'Edit Store: Default', 'Create A New Store', 'Save', and 'Cancel' buttons. Below the tabs, the 'Server' section contains the following options:

- Use SSL:** Radio buttons for 'Yes' and 'No'. The 'No' option is selected. A note below states: 'To use SSL check with your host if a SSL certificate is installed and added the SSL URL to the admin config file.'
- Encryption Key:** A text input field containing '12345'. A note below states: 'Please provide a secret key that will be used to encrypt private information when processing orders.'
- Use SEO URLs:** Radio buttons for 'Yes' and 'No'. The 'No' option is selected. A note below states: 'To use SEO URL's apache module mod-rewrite must be installed and you need to rename the htaccess.txt to .htaccess.'
- Output Compression Level:** A text input field containing '0'. A note below states: 'GZIP for more efficient transfer to requesting clients. Compression level must be between 0 - 9.'
- Display Errors:** Radio buttons for 'Yes' and 'No'. The 'Yes' option is selected.
- Log Errors:** Radio buttons for 'Yes' and 'No'. The 'Yes' option is selected.

We will see the available fields of server options.

- ▶ **Use SSL:** To use SSL check with our host, we need an SSL certificate installed and added to the SSL URL to the admin config file. For now, we will set it to **no**. There are many companies that provide an SSL certificate.
- ▶ **Encryption Key:** This encryption key will be used to process any private information securely. We will set it as **shops777**.
- ▶ **Use SEO URLs:** By choosing yes, our store will use SEO-friendly URLs.
- ▶ **Output Compression Level:** We can set the output compression level here. Nine is the maximum level.
- ▶ **Display Errors:** We will decide whether we want to display errors on the live site.
- ▶ **Log Errors:** We will log our errors.
- ▶ **Error Log Filename:** We set the error log filename.

We will save it with the **Save** button.

We have set the store configurations. Now, we will upload some products to our store.

Creating categories

We have set the configuration options for our store. Now, we will upload some products. But before that we will also create some categories for our products.

Getting started

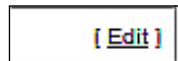
We will log in as admin with our admin username and password. Then, we will click the **Catalog** | **Categories** tab on the top navigation.



This will open up the following window with the default category list:

Category			Insert	Delete
<input type="checkbox"/>	Category Name	Sort Order	Action	
<input type="checkbox"/>	Desktops	1	[Edit]	
<input type="checkbox"/>	Desktops > PC	1	[Edit]	
<input type="checkbox"/>	Desktops > Mac	2	[Edit]	
<input type="checkbox"/>	Laptops & Notebooks	2	[Edit]	
<input type="checkbox"/>	Components	3	[Edit]	
<input type="checkbox"/>	Components > Mice and Trackballs	1	[Edit]	
<input type="checkbox"/>	Components > Monitors	1	[Edit]	
<input type="checkbox"/>	Components > Printers	1	[Edit]	
<input type="checkbox"/>	Components > Scanners	1	[Edit]	
<input type="checkbox"/>	Components > Web Cameras	1	[Edit]	
<input type="checkbox"/>	Software	4	[Edit]	
<input type="checkbox"/>	Phones & PDAs	5	[Edit]	

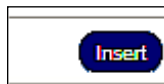
We have the **Edit** option available besides every category name.



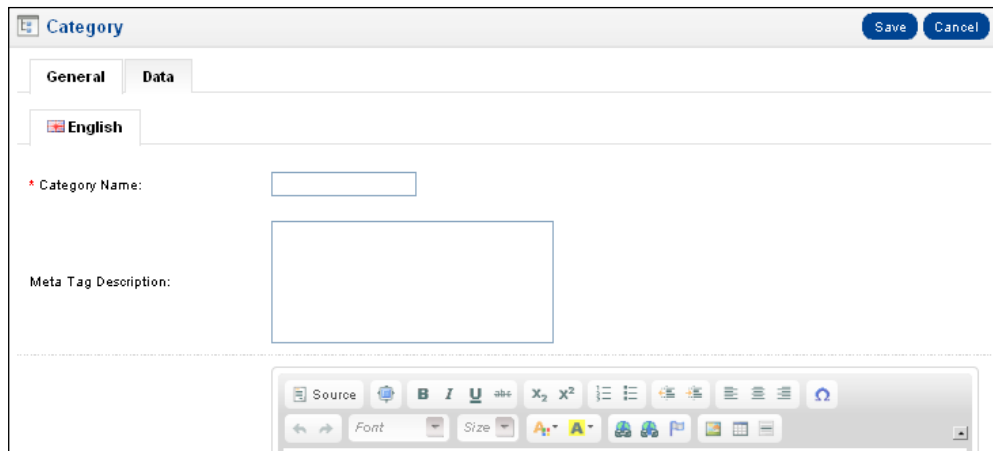
We can edit every existing category with this option. It will open up the edit window.

How to do it

Now, we will create some new categories for our store. For that we see an **Insert** button on the top right corner of the category list, like the following:



This will show the following category window for new category listing:

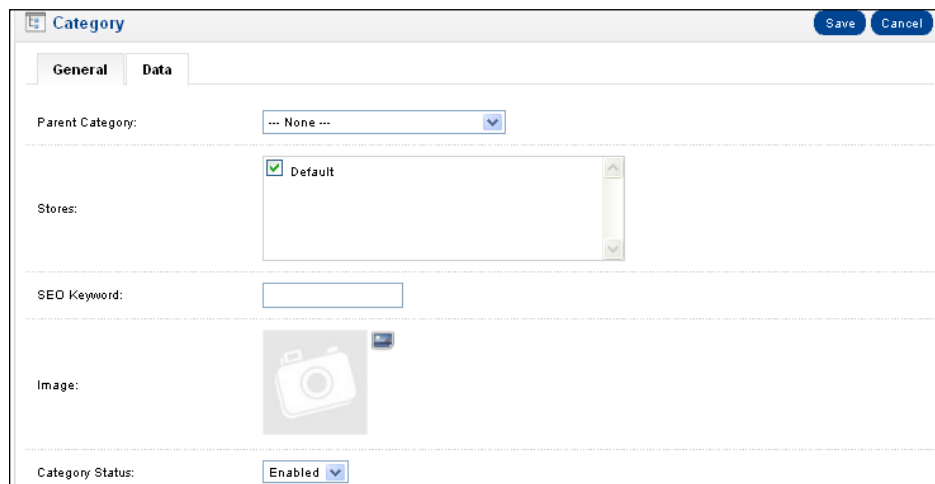


The screenshot shows a web application window titled 'Category'. It has two tabs: 'General' (selected) and 'Data'. Under the 'General' tab, there is a language selector set to 'English'. Below that, there are two text input fields: '* Category Name:' and 'Meta Tag Description:'. At the bottom of the window, there is a rich text editor toolbar with various icons for text formatting and editing.

This has two sections: general and data. We will populate them. But first, let us discuss the fields in them:

- ▶ **Category Name:** We will set our category name in this section. For example, we will set it as **cars**.
- ▶ **Meta Tag Description:** We will set some Meta tag description for our category. This will be helpful for our SEO. We will focus on appropriate keywords for writing this field. Also, try to write as many keywords as possible in this field.
- ▶ **Description:** This will be the description of our category. For now, we will set some dummy text from 'lorem ipsum'.

We have finished the **General** section. Now, we will fill up the **Data** section. We will click the **Data** tab:



The screenshot shows the same 'Category' window, but now the 'Data' tab is selected. The 'General' tab is still visible. The 'Data' section contains several fields: 'Parent Category:' with a dropdown menu showing '--- None ---'; 'Stores:' with a list box containing 'Default' and a checkmark; 'SEO Keyword:' with a text input field; 'Image:' with a placeholder image of a camera; and 'Category Status:' with a dropdown menu showing 'Enabled'.

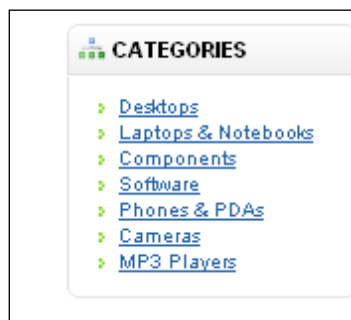
Following is the discussion of each field:

- ▶ **Parent Category:** As our category has no parent category, we will set it to **None**. For creating any subcategory, we can select a parent category from the available category list:



- ▶ **Stores:** We will leave it as it is.
- ▶ **SEO Keyword:** We will set appropriate SEO keywords for our category. For example, we will set it to **Car**.
- ▶ **Image:** We will upload our category image here.
- ▶ **Category Status:** We set our store status as **Enabled**. So, it will be available for display in the store front.
- ▶ **Sort Order:** We set it as 1. So, our category will be placed at the top of the category list. We can reorder our list at any time.

Now, we will save our new category by clicking the **Save** button. We will go to the store front to view our changes. Click the **Store Front** button on the top. We will see our newly created category at the top of the category list:



See also

We have created only one category. You can create lots more categories and sub-categories in the same way.

Adding products to a category







We have created a new category. Now, we will upload some products in that category. We will use the administration panel for this purpose.

Getting started

In this last recipe of this chapter, we need to log in as admin with our stored username and password.

How to do it

We will go to the **Catalog | Products** section. This will show the already available products on our store. We will see the following window:

Products							Insert	Copy	Delete
<input type="checkbox"/>	Image	Product Name	Model	Quantity	Status	Action			
		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="Filter"/>			
<input type="checkbox"/>		Apple Cinema 30"	Product 15	100	Enabled	[Edit]			
<input type="checkbox"/>		Canon EOS 5D	Product 3	10	Enabled	[Edit]			
<input type="checkbox"/>		HP LP3065	Product 21	10	Enabled	[Edit]			
<input type="checkbox"/>		HTC Touch HD	Product 1	10	Enabled	[Edit]			
<input type="checkbox"/>		MacBook	Product 16	9	Enabled	[Edit]			
<input type="checkbox"/>		MacBook Air	Product 17	10	Enabled	[Edit]			

Here, we can filter our products with product name, quantity, status, model attributes, and so on. See the option:

Product Name ▼	Model	Quantity	Status	Action
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text" value="▼"/>	<input type="button" value="Filter"/>

We also have **Edit** beside every product on the list. We can insert a new product. Also, we can make a copy of the existing products with the **Copy** button. By clicking **Insert**, we will be shown the following window:

Let's discuss the fields:

- ▶ **Product Name:** We will set our product name here. For example, we will set **aaa**. We can edit it at a later stage.
- ▶ **Meta Tag Description:** We will set some Meta tag description for our product. This will be helpful for our SEO. We will focus on appropriate keywords for writing this field. Also, try to write as many keywords as possible in this field.
- ▶ **Description:** We can write a description for our products. We have different styling properties available here.
- ▶ **Product Tags:** We can write comma-separated tags here for our products. For example, we will write **sport**.

Now, we will go to the **Data** section. It contains some important information about our products. Here, we have product model number, image upload options, SKU, product available date, Tax class, and so on.

Under the **Links** section, we can choose the manufactures, the category under which it will be listed, product download, and also, related products.

In the **Option** section, we can add different features for our product. Click the **Add Option** button to create new features.

We can add different discount values to our products. We can apply it to a select customer group and select when this discount will be available.

Like **Discount**, we can add **Special** price to our product. We will select a specific customer group for our product.

And in the **Image** section, we can upload an additional product image also. By clicking on the image, we can edit it.

We will save our product settings. So, our product has been uploaded and we can view it in our store front.

3

Layout Structure

In this chapter, we will cover:

- ▶ Creating a new theme
- ▶ Resetting layout styles
- ▶ Setting basic style properties
- ▶ Creating a promotional banner
- ▶ Placing a banner on your site

Introduction

In this chapter, we will see the default layout structure of OpenCart. We'll discuss the cascading stylesheet file from it and modify it according to our needs.

First of all, we'll use the reset style properties to eliminate cross-browser problems. We will discuss every property here. We'll also see the corresponding effects on our site.

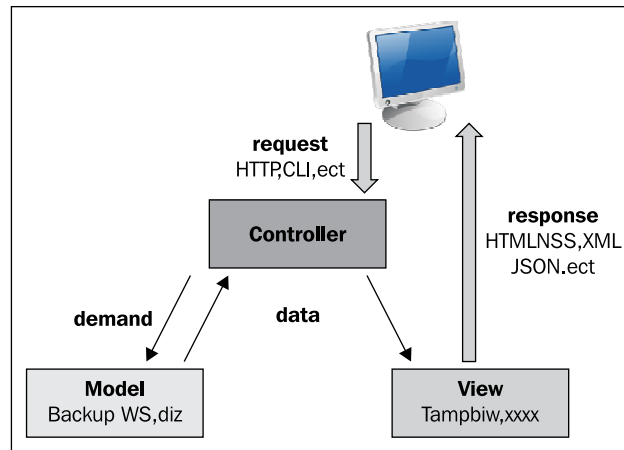
Then, we'll do some basic styling for the site. For each change in style, we will see why we did that.

Also, we'll make a banner image for our store in this chapter. This will be a step-by-step easy ride for the creation of our store's promotional banner.

Folder structure

In our main OpenCart folder, we have the `admin` and `catalog` sections. These are two separate subsystems. As the name says, `admin` has the files and folders for administration operation. `Catalog` contains the store files and folders. Each **admin** and **catalog** section has a separate model, view, and controller. Under this `admin` and `catalog` folder, we will see the model, view, and controller folders. You will see different subfolders within those folders. So, let's discuss this MVC structure in the following paragraph.

OpenCart is built with the MVC design pattern. So, it has model, view, and controller. A user requests the OpenCart controller to process the request. Then, the controller gets the data using the model and processes the fetched data to show the response with the view file. The following figure shows the above operation of MVC:



For theme modification, we will focus only on the `view` folder of the catalog in this chapter. It has `javascript` and `theme` folders. We place our themes under the `theme` folder and the necessary JavaScript files in the `JavaScript` folder.

Each theme has an `image`, `stylesheet`, and `template` folder. We will see how we can create a new theme later in this chapter.

Theme file style

As we stated earlier, OpenCart uses the MVC design pattern. So, the view files remain separated from the core code. These files are `.tpl` files. And, they are placed under `catalog\view\theme\default\template`. These `.tpl` files are basically HTML files. They have PHP code within them to display the necessary data.

OpenCart doesn't use the smarty template engine. Rather, it uses embedded PHP codes that are easy to use. We assign the PHP variables in the controller with necessary data. Then, we call the variable in the `.tpl` view file. We can also use the global class reference variable. In the controller, we will assign the value like this:

```
$this->data['shop_name'] = 'store';
```

Here, we assigned `store` value to the `shop_name` variable.

In the `.tpl` view file, we will display the value like this:

```
<?php echo $shop_name; ?>
```

Creating a new theme

In this recipe, we will see the steps to create a new theme for OpenCart. There are some rules to create OpenCart themes.

Getting started

Let's get started with the steps to create a new theme with OpenCart.

How to do it

Following are the steps for creating a new theme for OpenCart:

1. First of all, we need to create a new folder under `catalog\view\theme`. For example, we will name it `shop` for this project.
2. Now, we need to copy some files from the `default` theme folder to our new theme folder. The files are the following:
 - ❑ `catalog\view\theme\default\stylesheet*.*`
 - ❑ `catalog\view\theme\default\image*.*`
 - ❑ `catalog\view\theme\default\template\common\header.tpl`
3. We have to edit some values in the `header.tpl` file for our new theme. We will replace all `default` keywords with our new theme name `shop`. Actually, there are six places where we need to replace the new theme name. The lines are the following:

```
<link rel="stylesheet" type="text/css"
      href="catalog/view/theme/shop/stylesheet/stylesheet.css"
/>

// other lines ...
<link rel="stylesheet" type="text/css"
      href="catalog/view/theme/shop/stylesheet/ie6.css" />
//other lines ...
<div class="div3">
  <a href="<?php echo str_replace('&', '&amp;', $special);
?>"
    style="background-image: url('catalog/view/theme/shop/image/
special.png');">
<?php echo $text_special; ?></a>
<a onclick="bookmark(document.location, '<?php echo
addslashes($title); ?>');" style="background-image: url('catalog/
view/theme/shop/image/bookmark.png');">
```

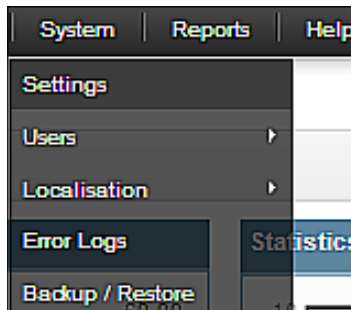


```
<?php echo $text_bookmark; ?></a><a href="<?php echo str_replace('&', '&amp;', $contact); ?>" style="background-image: url('catalog/view/theme/shop/image/contact.png');"><?php echo $text_contact; ?></a><a href="<?php echo str_replace('&', '&amp;', $sitemap); ?>" style="background-image: url('catalog/view/theme/shop/image/sitemap.png');"><?php echo $text_sitemap; ?></a></div>
//other lines ...
```

4. And now, save it.
5. Now, we will go to the admin area. First log in with our stored admin credentials.



6. Go to **System** | **Settings** in the admin panel:



7. We will go to the **Store** tab. We will change the theme from **default** to **shop**, our new theme from a select list.
8. You can make changes on your theme's CSS file.

Resetting layout styles

Before beginning our theme styling work, we must first reset all the styles. This will help us with cross-browser problems.

Getting started

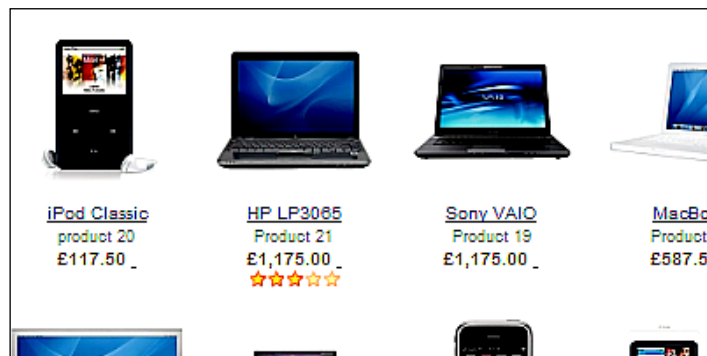
We need to modify the `stylesheet.css` file of our new theme first. We will go to `catalog\view\theme\shop\stylesheet`. And open up `stylesheet.css` in our favourite editor.

How to do it

1. Now, we will add reset styles to our `stylesheet.css` file. First, we need to change the browser's margin, padding, and border properties. We will set styles for several different HTML tags. We can add extra style properties into it:

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
    margin: 0;
    padding: 0;
    border: 0;
    outline: 0;
    font-size: 100%;
    vertical-align: baseline;
    background: transparent;
}
```

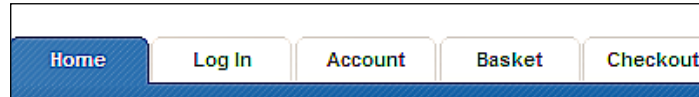
2. We will see the effect of our code in our store. The product images will come closer now.



3. We adjust the line height of body tag. So, put the following code in the CSS file .

```
body {  
    line-height: 1;  
}
```

4. This also squeezes the lines in the body element. The following image depicts this:



5. By applying the above style, the line height of these tabs becomes shortened.
6. We need to reset the style for ordered/unordered list elements. Hence, we use the following reset value:

```
ol, ul {  
    list-style: none;  
}
```

7. It shows all the ul, ol tags without the default bullet properties.
8. Now, we will reset the blockquote element styles. We will find the changes if we use blockquotes in our HTML code:

```
blockquote, q {  
    quotes: none;  
}  
  
blockquote:before, blockquote:after,  
q:before, q:after {  
    content: '';  
    content: none;  
}
```

9. For all the elements, we are going to change the focus-styling attributes. We change the outline properties to 0. We set the styles like the following:

```
:focus {  
    outline: 0;  
}
```

10. There could be some styling for insert and deletion in some browsers. So, we will use this styling for the purpose:

```
ins {  
    text-decoration: none;  
}
```

```
del {
  text-decoration: line-through;
}
```

11. We will control the styling of our tables also. We set the border and spacing qualities like the following:

```
table {
  border-collapse: collapse;
  border-spacing: 0;
}
```

12. We still need to set the attribute **cell-spacing** to 0.

13. So, our reset styling becomes the following:

```
html, body, div, span, applet, object, iframe,
h1, h2, h3, h4, h5, h6, p, blockquote, pre,
a, abbr, acronym, address, big, cite, code,
del, dfn, em, font, img, ins, kbd, q, s, samp,
small, strike, strong, sub, sup, tt, var,
b, u, i, center,
dl, dt, dd, ol, ul, li,
fieldset, form, label, legend,
table, caption, tbody, tfoot, thead, tr, th, td {
  margin: 0;
  padding: 0;
  border: 0;
  outline: 0;
  font-size: 100%;
  vertical-align: baseline;
  background: transparent;
}

body {
  line-height: 1;
}

ol, ul {
  list-style: none;
}

blockquote, q {
  quotes: none;
}
```

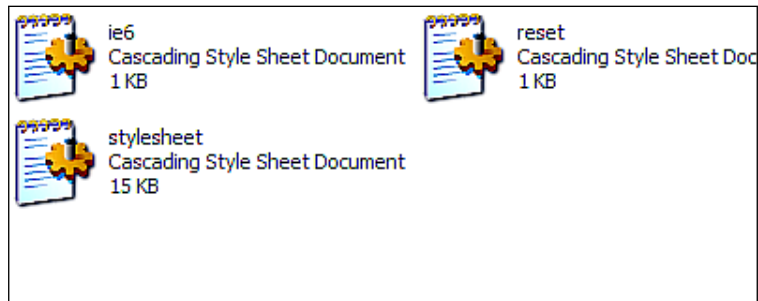
```
blockquote:before, blockquote:after,
q:before, q:after {
    content: '';
    content: none;
}

:focus {
    outline: 0;
}

ins {
    text-decoration: none;
}
del {
    text-decoration: line-through;
}

table {
    border-collapse: collapse;
    border-spacing: 0;
}
```

We can place it at the start of our site's style file `stylesheet.css`, or we can create a new file and put the content there also. To do that, just create a new CSS file within the `catalog\view\theme\shop\stylesheet` folder. For example, we can name the new style file as `reset.css`. If we use a new style file, then we need to add the style file in the controller.



Setting basic style properties

We have reset browser style properties. Now, we will apply our style to our site. There are some basic styling concerns.

Getting started

We are going to use the basic styling properties in this recipe. We need to modify the `stylesheet.css` file of our new theme first. We will go to `catalog\view\theme\shop\stylesheet` and open up `stylesheet.css` in our favourite editor.

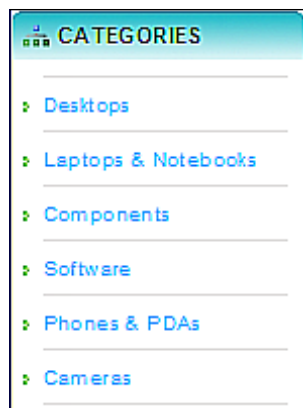
How to do it

We will have to do the following simple steps:

1. We need to set border properties and letter spacing for our site contents. For this, we use the following styling attributes. We will add it at the beginning of `stylesheet.css`.

```
html, body, div, span, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, address, code, del, dfn, em, img, q, dl, dt, dd,
ol, ul, li, table, caption, tbody, tfoot, thead, tr, th, td, br,
fieldset, textarea
{
border:0 none;
letter-spacing:0.5px;
}
```

2. We can use any convenient value for these properties. After setting the style, we will see the letters move slightly from each other.



3. Here, the letters are moved slightly from each other.

4. We can also add a value to our line height attributes. So, the code looks like the following:

```
html, body, div, span, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, address, code, del, dfn, em, img, q, dl, dt, dd,
ol, ul, li, table, caption, tbody, tfoot, thead, tr, th, td, br,
fieldset, textarea
{
border:0 none;
letter-spacing:0.5px;
line-height:15px;
}
```

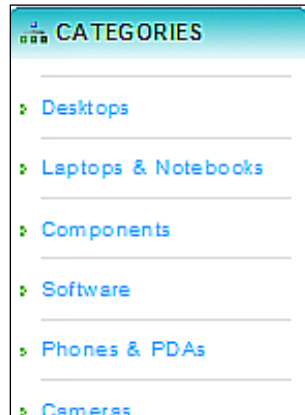
5. We have added 15px to the line height property. And this makes the following changes to our site:



6. You can easily find the difference by changing the value of this property. Each row now has a greater height.
7. Now, we can apply two other properties: **vertical-align** and **word-spacing**. See the following code block:

```
html, body, div, span, object, iframe, h1, h2, h3, h4, h5, h6, p,
blockquote, pre, address, code, del, dfn, em, img, q, dl, dt, dd,
ol, ul, li, table, caption, tbody, tfoot, thead, tr, th, td, br,
fieldset, textarea
{
border:0 none;
letter-spacing:0.5px;
line-height:20px;
vertical-align:baseline;
word-spacing:1px;
}
```

8. By applying word-spacing, each word of our site will move away from each other. We have applied 1 pixel here. You can make an adjustment of this value. The following image shows the spacing between two consecutive words:



9. We will format the heading text styles now. We will set font size, font weight, and colours of these properties. We will use the following code block for this purpose:

```
h1{
font-size:20px;
}

h2{
font-size:19px;
padding: 4px;
}

h3{
font-size:16px;
}

h4{
font-size:15px;
}

h5{
font-size:14px;
}

h6{
font-size:13px;
}
```

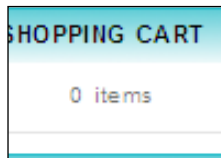

10. Here, we set the font size of some heading styling properties. We can adjust the font size according to our need. We set it for the overall site. But, it will not affect the styling of the main container as there is also a styling property available under the `content` ID. We will see how to change it. We need to change the heading text colours and the font weight now:

```
h1,h2,h3,h4,h5,h6{  
  color:#3f3f3f;  
  font-weight: normal;  
}
```

11. Our heading styles are set. Now, we will change the font colour for our site. Also, we'll set some other attributes for all over the site.
12. We first set our font colour to **#666666**.

```
body {  
  color:#666666;  
}
```

13. We can find the changes by visiting the site:



14. We also apply our font size for the site. So, the code block changes to this:

```
body {  
  color:#666666;  
  font-size:13px;  
}
```

15. Our changes will not be available into the site as there is also some styling in the default CSS that sets the font size.

```
body, td, th, input, textarea, select, a {  
  font-size: 12px;  
}
```

16. We remove this code from the stylesheet. So, the fonts get bigger now.



17. You have to add a style for the font family also:

```
body {
  color: #666666;
  font-size: 13px;
  font-family: Arial, Helvetica, sans-serif;
}
```

18. There is also another font styling for the overall site in the default CSS file:

```
* {
  font-family: Arial, Helvetica, sans-serif;
}
```

19. We will remove this code.

20. We add another property to the **body** tag. We set the text decoration property to **none**. So, the code block becomes like the following:

```
body {
  color: #666666;
  font-size: 13px;
  font-family: Arial, Helvetica, sans-serif;
  text-decoration: none;
}
```

21. We set a colour attribute to the **anchor** tag for our site.

```
a {
  color: #0066CC;
}
```

22. But the default style comes up with another styling for **anchor** tag.

```
a, a:visited {
  color: #1B57A3;
  text-decoration: underline;
  cursor: pointer;
}
```

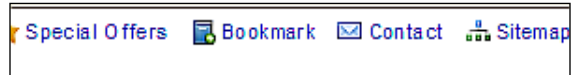
23. We remove this style. So, the style becomes like this:



24. We will add the following style to our **anchor** tag:

```
a {  
  color: #0066CC;  
  text-decoration: none;  
  outline: none;  
}
```

25. In the browser, we can see the changes like this:



26. For anchor-related styling, we will apply the **pointer** property to our cursor. See the following code:

```
a {  
  color: #0066CC;  
  text-decoration:  
  none; outline:none;  
  cursor: pointer;  
}
```

27. And now, we add a hover effect to our **anchor** style:

```
a:hover{  
  color: #CC6600;  
}
```



28. On hovering, the text color changes. There is a CSS styling property on the default CSS file.

```
a:hover {  
  text-decoration: none;  
}
```

29. We will remove it.

30. Now, we need to add some **margin** to some HTML elements:

```
h1, h3, h5, h6, dl, ol, ul, pre, table, address, fieldset {
    margin-bottom: 10px;
}
```

31. There will be a wide space after applying this style:

Price:	£1,175.00
Availability:	In Stock
Model:	Product 21
Manufacturer:	Hewlett-Packard
Average Rating:	★★★★☆
Qty: <input type="text" value="1"/> <input type="button" value="Add to Cart"/>	

Creating a promotional banner

In this recipe, we will see how to create a banner quite easily with GIMP for our store. We are going to see a step-by-step guide for it.

Getting started

We are going to place a banner image in the welcome message area. First, we will create the banner and then in the next recipe, we will place it using the admin panel.

How to do it

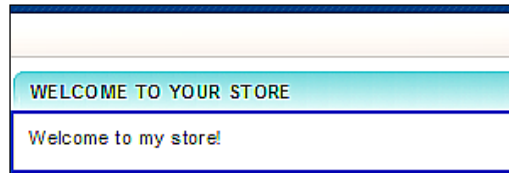
1. First of all, we need to measure the size of the welcome message area in order to have the image dimension. For this, click the **Firebug** icon:



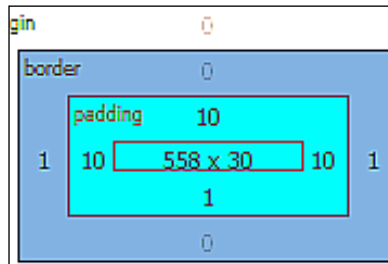
2. It will open up Firebug; now click on the firebug **inspector** icon:



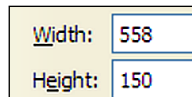
3. And click on the welcome message area on the browser screen. You'll see the following image:



4. If we click on the **layout** option on the right side of the Firebug option, we can find the dimension of the selected area:



5. Actually, we have got only the width of our banner image from this information. We will adjust the height accordingly.
6. Now open GIMP. Press **Ctrl+N** to create a new image. Set the dimension as 558x150. We use the width and make a height according to our need.



7. We will give a **background** colour to our banner image. Select the **color** tool from the toolbox.



8. You can choose your favourite colour from this colour panel. We choose **#7c1313** for this recipe.

9. Now, click the **Bucket** Tool from the toolbox:



10. Now, pour your selected colour on the banner image area. I have chosen the following color (maroon):



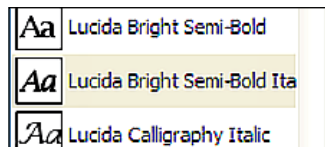
11. We are going to write some promotional text on our banner image. Select the **Text** Tool from the toolbox:



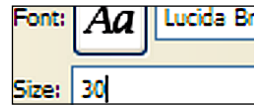
12. We will select our font style from the bottom text dialog box of GIMP. You can choose font styles by clicking on the **Font** button at the dialog box:



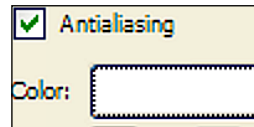
13. There are a wide range of font styles. So, you can choose your favourite one from there. We will choose **Lucida Bright Semi-Bold Italic** for now.



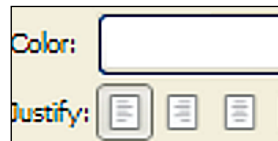
14. And set the size to **30px**:



15. Choose the font colour #ffffff.



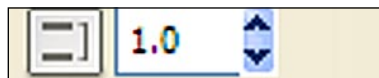
16. We write our text as left aligned. So, select **justify** option as left aligned:



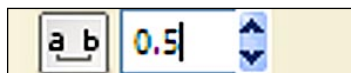
17. We leave indentation to **0px**:



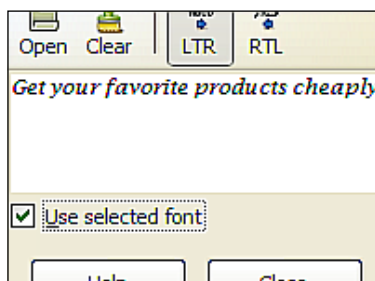
18. The line height of our text will be **1px**:



19. And letter spacing will be **0.5px**:



20. Now, select the rectangular area in your image that you want to write on. A new window might have opened for writing the texts. We can write our texts there.



21. We can set our text as left aligned or right aligned. After writing, click the close button.
22. You can see a rectangular area around our text. By clicking the corners, you can adjust the dimension of the rectangular text area. At the moment, our banner becomes like the following:

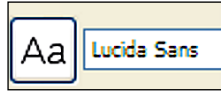


23. Now, we write a subtext below the above text in our banner. We will use a different font style to write this. But you can use the same style if you wish.
24. You can see currently our text selection area covers most of the right side of the image. To write a subtext, we need to shorten the height of the text area.
25. Click and drag the corner boxes of the bottom area upward. This will reduce the height. See the following image:

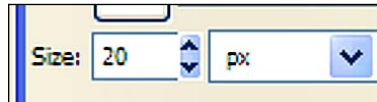


26. Here, we reduce the height.

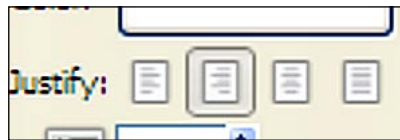
27. Now, for subtext, we will change our font style to **Lucida Sans**:



28. We need to change the font size also. We will change it to **20px**:



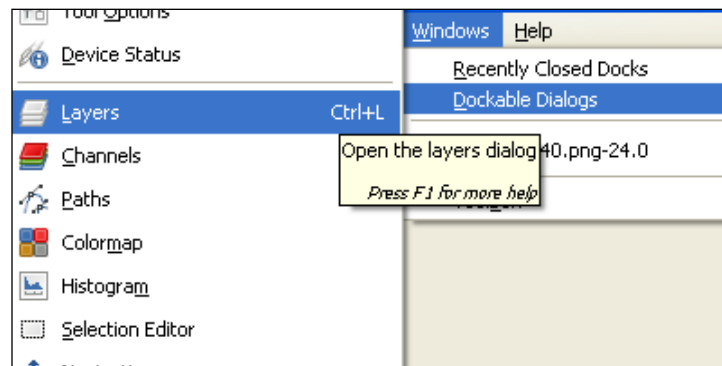
29. We right aligned our subtext. So, change the **justify** option to right aligned:



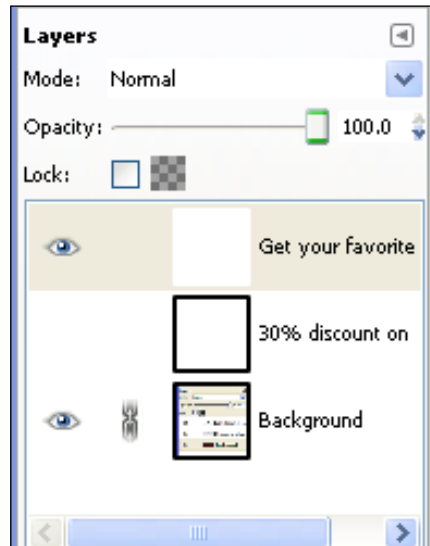
30. Now, select an area below the current text on the banner to write subtext. Our banner image will become like the following:



31. We will use different brushes to create some effect. First, go to **Windows | Dockable dialogs | Layers**:



32. We will see that the following layers window opens:



33. We have three layers for our banner. We will use brushes on the background layer. So, we hide the other layers by clicking the **eye** icon. Now, we have only the background layer visible.

34. Click on the brush icon on the toolbox:

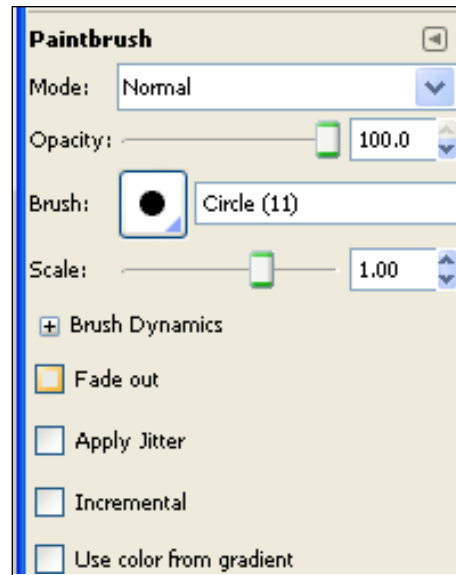


35. It will open up the paintbrush properties dialog.

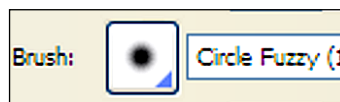
36. Let's discuss the properties of the paintbrush tool:

- ❑ **Mode:** We select the mode for our paintbrush from the dropdown.
- ❑ **Opacity:** The opacity of the paint brush can be controlled from this sliding option.
- ❑ **Brush:** There are a wide range of brushes available. Also, you can create your own or you can download and install a new brush.
- ❑ **Scale:** We can scale our brush with this sliding option.
- ❑ **Brush Dynamics:** There are many more options under this option. We will see them one by one here.

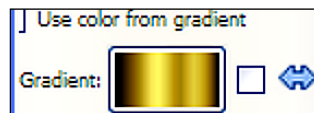
- ❑ **Pressure, velocity, randomness:** we can make a combination of these values with opacity, hardness, size and color.
- ❑ **Fade out:** by selecting the check box, there will be a sliding option to select the length. The brush will be faded out after the selected length.
- ❑ **Apply jitter:** we can control jitter amount with this brush.
- ❑ **Use color from gradient:** we can choose the color of our brush from some gradients. We set the length and repeat type.



37. We will select our **brush** from the brush dialog box. We select the **brush Circle Fuzzy (17)** brush:



38. Under the brush dynamics, you can see many different options. You can use them according to your need. We choose the following gradient for our brush:



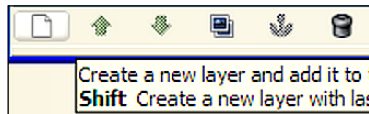
39. With the selected gradient, we can draw something on the background using the brush. We have made the following banner image up to this stage:



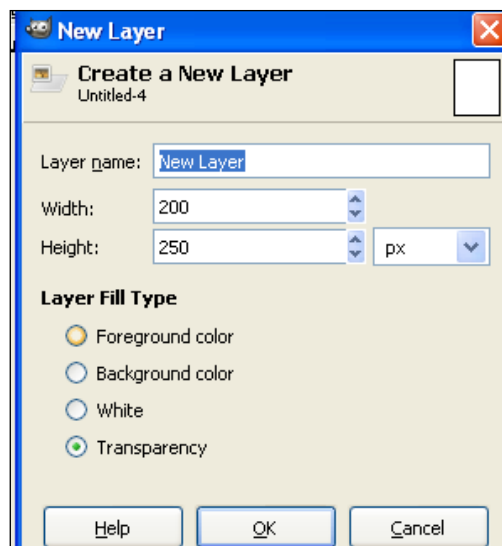
40. Now, if we show the other two text layers, then we see the banner as:



41. Our banner has left-side empty. We will place an image on the left side. We can search the Internet for an appropriate image.
42. First, we will add a new layer to our banner. Click on the **new layer** button at the bottom of the layer dialog:



43. It will open the following dialog for creating a new layer:



44. The following are the fields of the layer tool:

- ❑ **Layer name:** We will set the layer name here. For example, we will set it **banner image**.
- ❑ **Width:** The width of the layer. We set it 558px.
- ❑ **Height:** The height of the layer. We make it 150px.
- ❑ **Fill type:** We make it **transparency**.

45. For this image, we will add an image of a shopping cart and a person.

46. We use **fuzzy select tool** from the toolbox:



47. Crop the image and place it on the left side of the layer. So, finally, our banner becomes the following:



Placing a banner on your site

Here, we will place our newly created banner on our site using the admin panel.

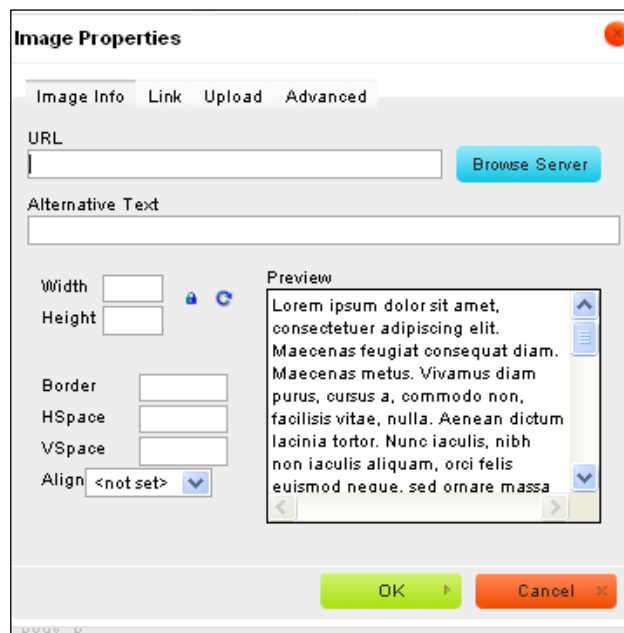
Getting started

We need to log in to our admin panel. We will use our stored credentials for login. After logging in, we will upload the banner.

How to do it

Let's do the following steps:

1. Go to **System | Settings**. Now, click on the **Store** tab.
2. Under the **Welcome message** area, click on the **Image** icon from the editor option panel.
3. Upload your image, using the **Upload** tab. And click **Ok**.
4. Now, on the image info tab, browse to your uploaded image. Set the height and width. We will set to it 150px and 558px respectively.



5. Click **OK**.
6. Click the **save** button on the top right corner of admin panel.
7. Go to the store front to view the banner.

4

Module Adjustment

In this chapter, we will cover:

- ▶ How to change the position of a module
- ▶ Creating a header image for the module block
- ▶ Applying styles to each block element of a module
- ▶ Changing the category block
- ▶ Modifying the featured block

Introduction

As the title says, we will see the OpenCart modules and their structures in this chapter. So far, we have set up our shop and configured it properly. In this chapter, our focus will be on module adjustment. We can place some modules on different positions of the store front.

So, they have a different layout structure for the same module depending on their position. Also, the default layout structure for the OpenCart module is table-based. So, we will change the structure to div-based.

In this chapter, we focus on the adjustment of the cascading stylesheet properties for our modules. We use GIMP and Firebug to do this styling.

Our first recipe is to know our OpenCart modules. We will examine the existing modules and their structures. We will also see some other helpful modules.

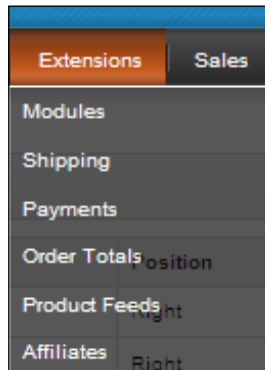
We are going to see how we can change the position of the modules and also different structures of the same module in different positions.

Getting ready

For module installation, we need to go to our back office or admin. We will use our stored login credentials.

How to do it...

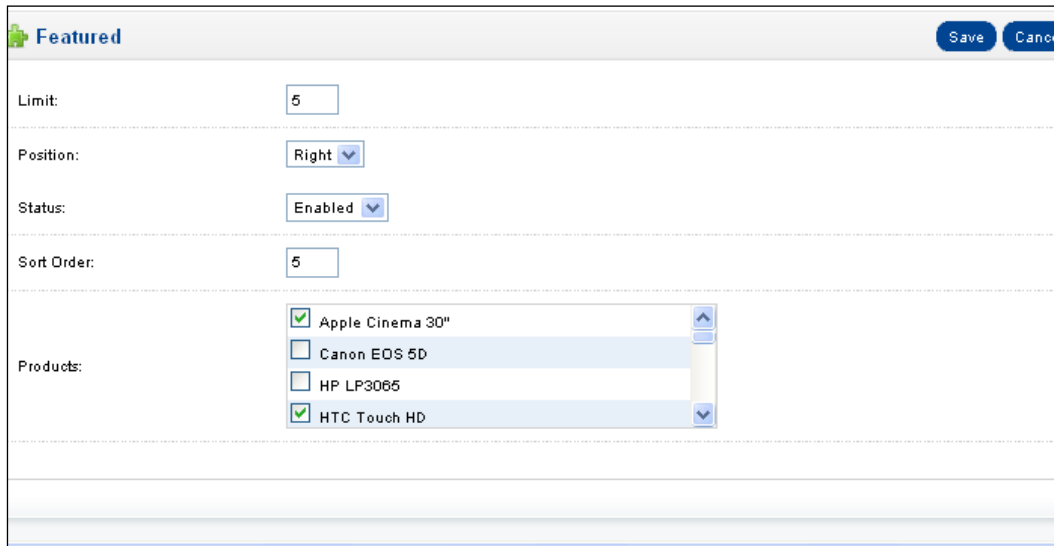
1. In the admin, we will go to **Extension | Modules**.



Then, we will see all the available modules of our store. It shows the status of the module with module position. We can edit each module. The following window shows the modules:

Modules				
Module Name	Position	Status	Sort Order	Action
Bestsellers	Right	Enabled	3	[Edit] [Uninstall]
Cart	Right	Enabled	1	[Edit] [Uninstall]
Category	Left	Enabled	1	[Edit] [Uninstall]
Featured	Right	Enabled	5	[Edit] [Uninstall]
Google Analytics		Disabled		[Install]
Google Talk		Disabled		[Install]
Information	Left	Enabled	3	[Edit] [Uninstall]
Latest		Disabled		[Install]
Manufacturer	Left	Enabled	2	[Edit] [Uninstall]
Specials		Disabled		[Install]

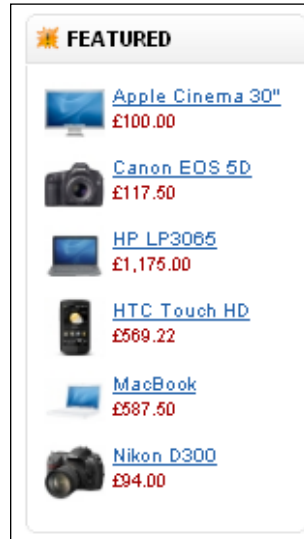
- There is the module name, position, status, sort order, and action. Under the action, we have the edit, install, and uninstall options. Let's edit one of our available modules. We are going to edit the **Featured** modules. Click on the **Edit** button. This will open up the following window:



Let's discuss the fields of the **Featured** module:

- ❑ **Limit:** This controls the number of products we want to show in the featured block of our store front. We will set it to six.
- ❑ **Position:** We can place it on left, right, or at the middle of the homepage. But some modules cannot be placed at the middle of the homepage, for now, we will place it on the left side.
- ❑ **Status:** We may set the **Enabled** or **Disabled** option. To show our module, we need to enable it.
- ❑ **Sort order:** With this, we can control the placement of the module. The less the sort order, upper will be the position of the module. On the left column, we have our modules: category, manufacturer, information, featured with the sort order of 1, 2, 3, and 5 respectively. If we want to show our module on the upper position, we need to readjust our module position; it may be like this: category, manufacturer, information, featured may be 2, 3, 4, and 1 respectively. Now, our featured module will be placed at the top of the left column.
- ❑ **Products:** We will select our featured products from this list. We will choose the same number of products that we set as limit value. We will choose six products.

We will save our changes. Now our featured module will be like this:



Changing header of module

We have placed our featured module on the left side of our store front. It has the default layout structure. In this recipe, we are going to change its layout structure.

Getting started

To change the layout structure, we will go to the OpenCart folder of our document root, that is, `Xampp\htdocs`. Then, we will go to our project folder. We will copy the module folder of the default template to `catalog/view/theme/shop/template`. In the `catalog/view/theme/shop/template/module`, we will see `feature.tpl` file. This one is for left/right column and if we set featured products on the middle of the homepage, there will be another file named `feature-home.tpl`. For now, we will change the featured `tpl` file for the left/right column.

How to do it...

1. In the Featured `tpl` file, this block is within a class of 'box'. It is divided into three sections: top, middle, and bottom. Top section is for header of the block, middle div contains the body, and bottom is the bottom layer.
2. In the top class, we have code like this:

```
<div class="top"><?php echo $heading_title; ?></div>
```

- The image tag contains an icon for featured header block. We can create a new icon and replace it. We can search for a featured icon on the Internet. Suppose, we use the following for our featured products block:



- We are going to edit this image. So, let's open this image with GIMP. As the icon is too small to work with, we will enlarge the image with the **zoom** tool:



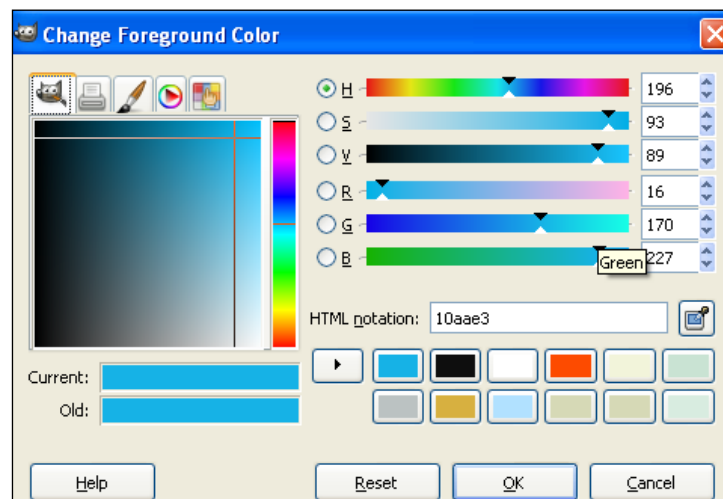
- By clicking the left button of the mouse on the image, we can enlarge it according to our need. Now, we are going to change the image color. First, we will change the outer ring of the image. For this, select the **select by color tool** on the tool box.



- Then, we will select the outer ring with this tool. Now, we will choose a proper color from the color tool:



- We will see the color panel. For the featured block, we will choose **#10aae3**:



8. So, our color is now **#10aae3**. And now, we will choose the **Gradient** tool from the tool box and apply it in our selected image region. Now, our images will be like this:



9. We can change the font color to white. Choose the **Fuzzy select tool**:



10. And now, select the font area on the image. We will change our color to **#ffffff**. Now, choose the **gradient** tool and apply the color on the select area. So, our icon will be like this:

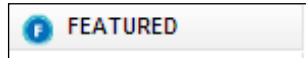


11. We will save it under the `catalog\view\theme\shop\image`. We have our icon and now we will change the image name on the **top** block. In the featured tpl file, we will place our image. We have used the `Featured-icon-name.png` just for example, you can use other names.

```

```

12. We will see our icon in action. Go to the store front and refresh the browser:



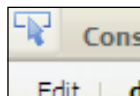
13. If we like, we may remove icons from the top block. For this, we need to just remove the image tag. Then, our top block will become like this:

```
<div class="top"><?php echo $heading_title; ?></div>
```

14. We can change the top header color also. Let's see how we can change that. We are going to use Firebug for this. Click on the **Firebug** icon on the bottom right corner.



15. Our Firebug window will open, click the **Inspect** icon to inspect HTML:



16. Click the Inspect icon and catch the top header block on the browser. Then, the relevant code will be viewed on the Firebug console. Click on the **box** class and corresponding styles will be shown on the **style** tab on the right panel. You will find the background image for the top header block.
17. We are going to change the image of the top header. The image is on the image folder of the **shop** theme. We will open up the image **box_top** with GIMP.
18. Click the **select by color tool** from the toolbox. Now, click on the main center region of the image. The area will remain selected. Now, click the color tool. And choose the color we want to apply to our header. We will choose #11b2c1. Now, choose the **Gradient** tool. We apply it from top to bottom:

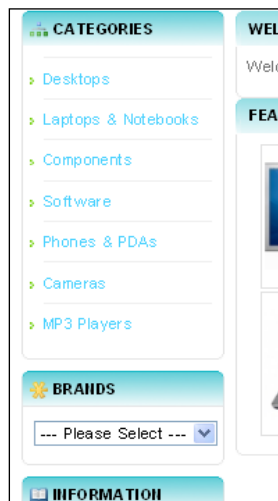


19. If you zoom into the image, you will see there is a border around the image. We will change the color also. First select the **fuzzy select tool** from the tool set. Click on the edges. Then, choose the **bucket fill tool** and click on the selected edge. Now, the image is changed to the following:



20. If we go to the store front and refresh our browser, we can easily view our new background image. We can change the image name by going to the stylesheet .css file of our theme. Go to line number 295, we will see this code:

```
background: url('../image/box_top.png') no-repeat;
```
21. We can change the image name here. Now, let's see how it is looking in the browser:



Adjusting layout

In this recipe, we will adjust the layout structure of the featured block. We will apply different style properties to the HTML tags. Currently, the block is table-based and we will convert it to div-based structure.

Getting started

In the featured tpl file, the default layout is table-based. We will make it div based. For that, notice the featured products are showing in a list. We can use **ul li** tags for this purpose.

How to do it...

1. First, we will remove the table tags. By table tags I mean table, tr, td. Now, you will see that there is a `foreach` loop to print out the featured products. We will place the `ul` tag to wrap the `foreach` loop.

```
<ul id="featured">
    <?php foreach ($products as $product) { ?>
    // ...
    <?php } ?>
</ul>
```

2. Here `// ...` means other codes. We named the `id` attribute as `featured`.
3. And now, place the `li` tag within the `foreach` loop; just after the start of the loop.

```
<?php foreach ($products as $product) { ?>
    <li>
    // ...
    </li>
    <?php } ?>
```

4. We want to show the product image on the right side and its description on the left side. For that we bring the image tag just after the product description. We will wrap the image tag with `span` and the product description with another `span`; we can also write styles on the `stylesheets.css`; like the following:

```
<span class="featured-description"><a href="<?php echo str_
replace('&', '&amp;', $product['href']); ?>"><?php echo
$product['name']; ?></a>
    <?php if ($display_price) { ?>
    <br />
    <?php if (!$product['special']) { ?>
    <span style="font-size: 11px; color: #900;"><?php echo
$product['price']; ?></span>
```

```

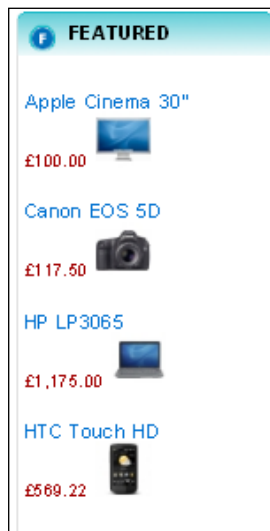
        <?php } else { ?>
        <span style="font-size: 11px; color: #900; text-
        decoration: line-through;"><?php echo $product['price']; ?></
        span> <span style="font-size: 11px; color: #F00;"><?php echo
        $product['special']; ?></span>
        <?php } ?>
        <?php } ?></span><span class="featured-image"><a
        href="<?php echo str_replace('&', '&amp;', $product['href']);
        ?>">"
        /></a></span>

```

5. Now, we will apply some styles to our feature block. Go to `catalog\view\theme\shop\stylesheet` and open up our css file `stylesheet.css`. We will add the style attribute here.
6. For the **featured** ID attribute, we will add some padding to the style. See the following code block:


```
#featured { padding : 5px; }
```
7. Also, we will apply the following styles in the `li` tag:


```
#featured li {padding: 10px 0px; }
```
8. At this stage, our featured block becomes like this:



9. We will show product description on the left side and image on the right side. So, now, we will apply the following style to the span class featured-description and featured-image:

```
.featured-description {float: left; padding-right: 5px;}  
.featured-image img{border: 1px solid #ddd; padding: 3px;}
```

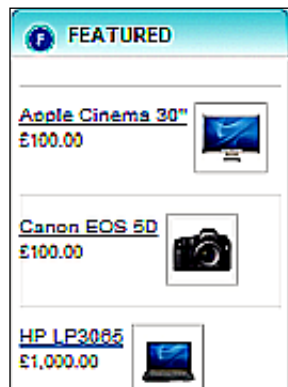
10. Now, our block becomes like this:



11. On mouse over, we are going to change the style like this:

```
#featured li:hover { background-color: #f7f7f7;}
```

12. And it becomes the following when we hover the block. It has a slight gray background color:



13. Now, we will change our text decoration. Also, change text color and hover text effect. Use the following code for this:

```
#featured li a { text-decoration: none; color: #3FCDFD; }  
#featured li a:hover { color: #FFC62F; }
```

14. You can change the text color to your favorite one and also the text hover effect. Our feature block now becomes the following:



15. When we hover our text links, it becomes like this:



16. We may also add some hover style to our featured product image. If we can enlarge our image with some css style, then it will be more eye catching to our visitors. We will apply the following style:

```
.featured-image img:hover{ border: 1px solid #BFBFBF; padding: 7px; }
```

17. So, when we hover on the product image, it will become larger:



18. So, our complete css stylesheet for the featured block is like the following:

```
/* ---- start of featured block ----*/
#featured { list-style-type: none; padding: 0px; margin: 0px;
padding : 5px;}
#featured li { padding: 10px 0px; border-top: 1px solid #eee;}
#featured li:hover { background-color: #f7f7f7;}
#featured li a { text-decoration: none; color: #3FCDFE; }
#featured li a:hover { color: #FFC62F; }
.featured-description { float: left; padding-right: 5px;}
.featured-image img{ border: 1px solid #ddd; padding: 3px;}
.featured-image img:hover{ border: 1px solid #BFBFBF; padding:
7px;}
/* ---- end of featured block ----*/
```

See also

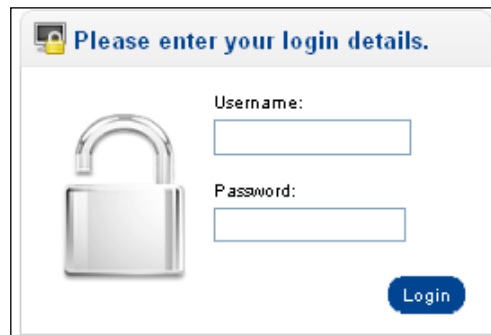
There are many more modules available in OpenCart. We can adjust those layouts also. But not all modules can be placed on the left and right columns.

Placing featured block at middle

We have seen the featured block on the left column. It will have the same attributes if we place it on the right column also. Featured block can also be placed in the middle column. But then, the layout of the featured block will be different.

Getting started

In the backend of OpenCart, we can change the module position. We need to log in to the admin panel:



Use our stored credentials for admin login. Then, we will change the position of the featured block to the middle section.

How to do it...

1. In the admin panel, go to **Extensions | Modules**. All the available modules will be shown in the window. We will see the featured block at the left position.

Bestsellers	Right	Enabled	3	[Edit]
Cart	Right	Enabled	1	[Edit]
Category	Left	Enabled	1	[Edit]
Featured	Left	Enabled	5	[Edit]

2. Click the **Edit** button to edit the featured block content. We will see that the position attribute is set to **left**. We will change it to **Home**:

Position: Home ▼

3. We can change the number of products with the **Limit** option. For now, we will set it to **4**. See the following figure:

Limit: 4

4. And accordingly, we will select the same number of products from the **Products** multiple select area:

Products:

- ☒ Apple Cinema 30"
- ☒ Canon EOS 5D
- ☒ HP LP3065
- ☒ HTC Touch HD

5. We have chosen four products from the list.
6. We may adjust the sort order of the center module blocks. But for now, we will place it at the bottom of the center column on the homepage. So, we place **2** to this field. See the following figure:

Sort Order: 2

- Now, we will save our changes by clicking the **Save** button. We will see our magic by going to the store front. You can see that our featured block is placed at the middle column on our store homepage.



- In the next recipe, we are going to change its layout style.

See also

Here, we have only shown the placement of the featured block. We can also change the position of the following blocks:

- ▶ Latest products
- ▶ Best seller
- ▶ Special products

Styling centre featured block

Our featured product module is now at the centre column. It has default layout attributes. We have changed the left positioned featured block. Now, in this recipe, we will focus on enhancing the style of the middle positioned featured block.

Getting started

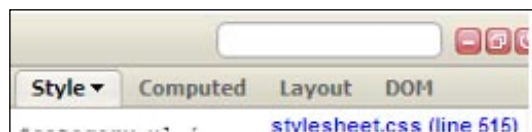
We have already placed our featured block at the middle position. So, we don't need to go to the admin panel. First, we examine the block with Firebug.

How to do it...

1. The centre column featured block header needs to match with our site's other headers. It is currently showing the old style. So, we will change it. Select the **Firebug** icon, and inspect it. If we place our inspector on the featured block header, it will show the following code in the console:

```
<div class="bottom">
  <div class="top">
    <div class="left"></div>
  <div class="right"></div>
  <div class="center">
    <h1>Featured</h1>
  </div>
</div>
<div class="middle">
```

2. Now, click on the **center** class div, we will find the image to change. It is `content_top_center.png`:

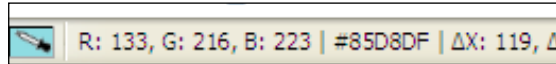


```
#content .top
.center{
  -moz-background-clip: border;
  -moz-background-origin: padding;
  -moz-background-size: auto auto;
  background-attachment: scroll;
  background-color: transparent;
  background-image: url("../image/content_top_center.png");
  background-repeat: repeat-x;
}
```

3. Now, open the image with GIMP to edit it and enlarge the image for editing purposes. Now, click on the **color** tool on the toolbox.
4. We find the already used side module block header color with **Colorzilla**. Click on the **Colorzilla** icon:



5. You need to move the mouse on the left-side module header and click on the color. Then, the Colorzilla icon will change to our selected color. Also, it will show the color attributes and positions:



6. We have got our color code: #85d8df. Now, we will apply it to the top image of the module. We will choose the color code from the already open color panel of GIMP. Then, the **Color** tool will change to this:



7. Now, select the **Gradient** tool from toolbox. You need to apply the color from top to bottom on the image. Our image will become like this:



8. There is also left and right side image of this top header of the module. We will find it in the same way. The images are:

- ☐ content_top_left.png
- ☐ content_top_right.png

9. We modify those images in the same way. We need to use the **Fuzzy Select Tool** and the **Bucket Fill Tool** from the tool box for this purpose. The left image will become like this:



10. And the right one will look like this:



11. If we see our store front, we can view the changes there:



12. Now, we will modify the code. We find the code at `featured_home.tpl` under `catalog\view\theme\shop\template\module`. Currently, it is table-based. But we will convert it to div-based. The following is the table-based code:

```
<div class="middle">
  <table class="list">
    <?php for ($i = 0; $i < sizeof($products); $i = $i + 4) { ?>
      <tr>
        <?php for ($j = $i; $j < ($i + 4); $j++) { ?>
          <td style="width: 25%;"><?php if (isset($products[$j])) { ?>
            <a href="<?php echo str_replace('&', '&amp;',
$products[$j]['href']); ?>">" alt="<?php echo $products[$j]['name'];
?>" /></a><br />
            <a href="<?php echo str_replace('&', '&amp;',
$products[$j]['href']); ?>"><?php echo $products[$j]['name']; ?></
a><br />
            <span style="color: #999; font-size: 11px;"><?php echo
$products[$j]['model']; ?></span><br />
            <?php if ($display_price) { ?>
              <?php if (!$products[$j]['special']) { ?>
                <span style="color: #900; font-weight: bold;"><?php echo
$products[$j]['price']; ?></span>
              <?php } else { ?>
                <span style="color: #900; font-weight: bold; text-
decoration: line-through;"><?php echo $products[$j]['price']; ?></
span> <span style="color: #F00;"><?php echo $products[$j]['special
']; ?></span>
              <?php } ?>
            <?php } ?>
          <?php } ?>
        <?php } ?>
      </tr>
    </tr>
  </table>
</div>
```



```
        <a class="button_add_small" href="<?php echo
$products[$j]['add']; ?>" title="<?php echo $button_add_to_cart;
?>" >&nbsp;</a>
        <br />
        <?php if ($products[$j]['rating']) { ?>
        " />
        <?php } ?>
        <?php } ?></td>
        <?php } ?>
    </tr>
    <?php } ?>
</table>
</div>
```

13. We will remove the table and use `ul` `li` tags instead. Place the `ul` tag in place of table tags. Like the following:

```
<ul id="featured-middle">
    <?php for ($i = 0; $i < sizeof($products); $i = $i + 4) { ?>
    // ... internal code block
    <?php } ?>
</ul>
```

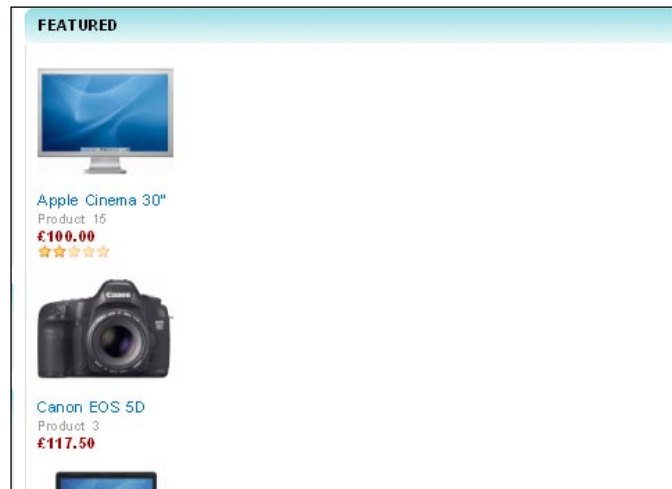
14. Now, remove the `tr` and also `td`. And in place of `td` we will place `li`. We have shown the code block below:

```
<li><?php if (isset($products[$j])) { ?>
// ... internal code block
<?php } ?></li>
```

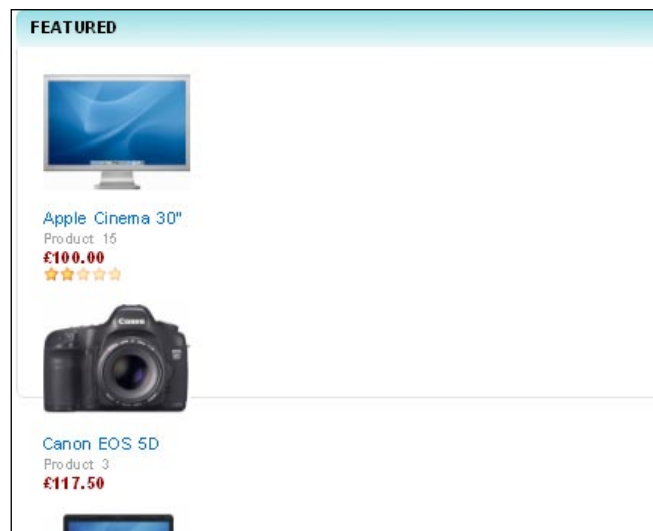
15. We will divide the content within the `li` tag into two sections. We use `div` for this purpose. One section is for product image and another section is product description. We wrap those two sections with `div` tag and add class `product-image` and `product-description`. See the following code block.

```
// ... product image
<div class="product-image"><a href="<?php echo str_
replace('&', '&amp;', $products[$j]['href']); ?>">" alt="<?php echo $products[$j]['name'];
?>" /></a></div>
// ... product description
<div class="product-description">
    <a href="<?php echo str_replace('&', '&amp;',
$products[$j]['href']); ?>">
    // ... internal code
    <?php } ?></div>
```

16. If we see our featured block in the browser, we find the following status of our module:



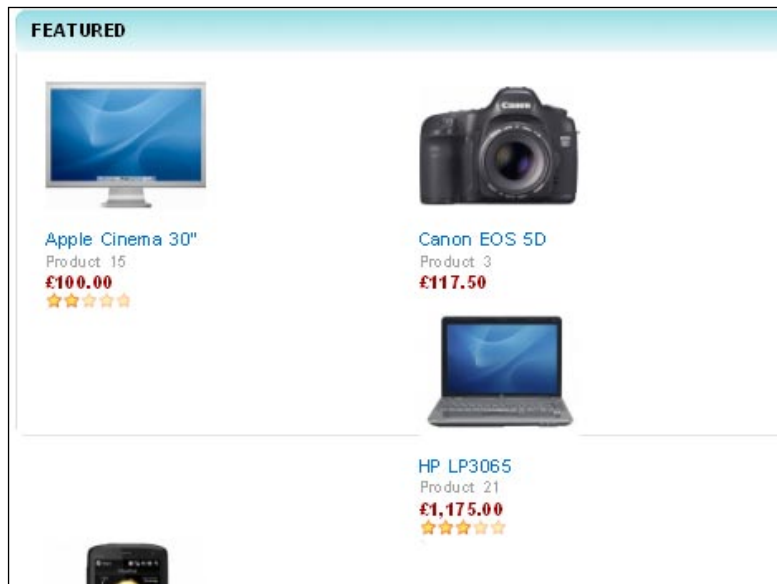
17. Now, it's time to apply css to our module block. Go to `catalog\view\theme\shop\stylesheet` and open up our site's stylesheet.
18. We will apply the following code to our **featured middle** attribute of the `ul` tag:
- ```
#featured-middle {padding: 0px 0px 0px 10px; list-style-type: none; width: 100%; height: 260px;}
```
19. This will change our module block to the following status:



20. Our products are showing on the one line. So, you need to use the following code in the `li` tag. See the code block:

```
#featured-middle li {width: 50%; float:left; display: inline;}
```

21. This changes the featured module to this stage:



22. Our product image will be shown in the left and its description on the right. We use the following code block:

```
#featured-middle .product-image {float: left;}
```

23. This makes our module block look like the following image:



24. You see that our product description is very adjacent to our product image. So, we need to apply some style to move it to the right. You can use the following code for this:

```
#featured-middle .product-description {float:right; padding: 20px 30px 0px 0px;}
```

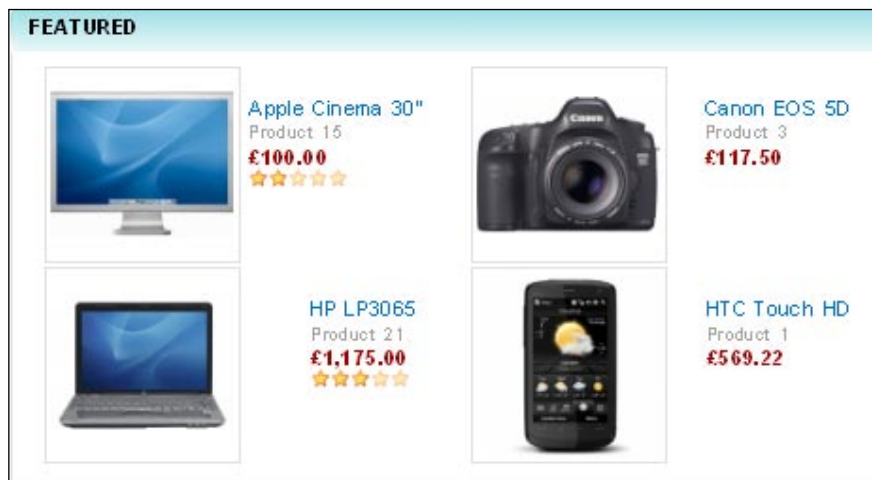
25. And our module becomes:



26. We use some styles to give a border around the product image. You may use the following code block:

```
#featured-middle .product-image img{ border: 1px solid #ddd; padding: 3px;}
```

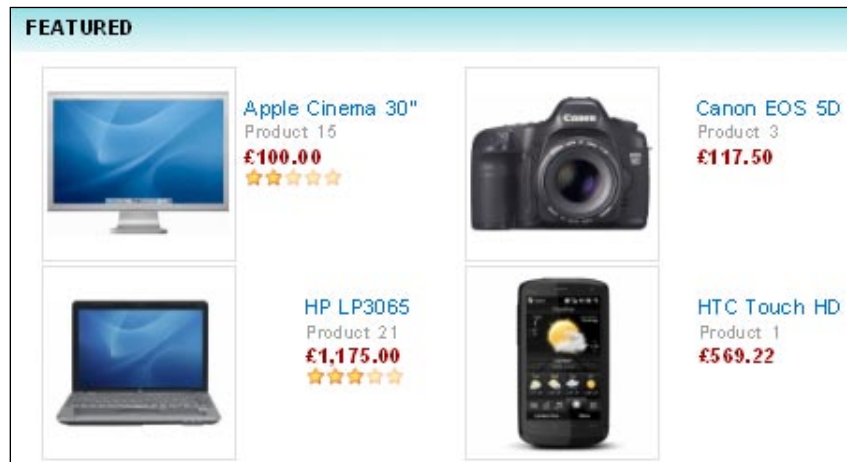
27. So, we can view the borders around our image now:



28. When we hover over a featured product block, we can add some style to it. We can write the following code:

```
#featured-middle li: hover { background-color: #f7f7f7; }
```

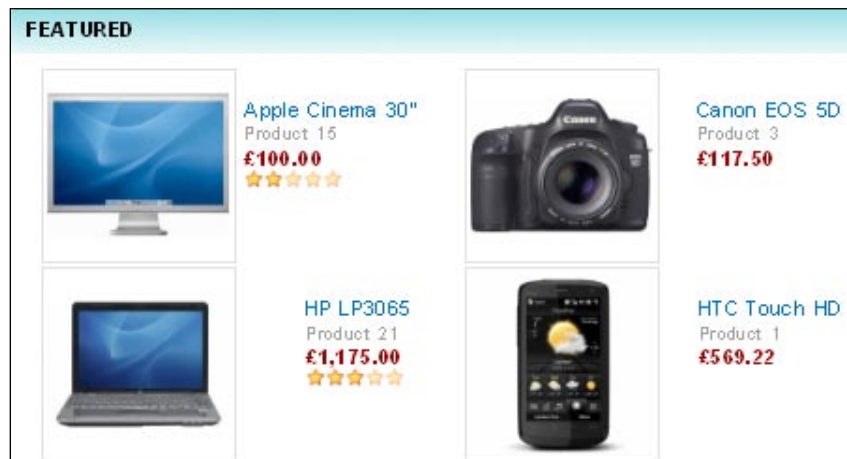
29. See the changes on the browser:



30. You should use same text style for every module with some exceptions. So, we make the following styling code for the text style:

```
#featured-middle li a { text-decoration: none; color: #3FCDFE; }
```

31. This gives the following changes in our browser:



32. Apply some hover effect on the text. You can use the following hover style:

```
#featured-middle li a:hover { color: #FFC62F; }
```

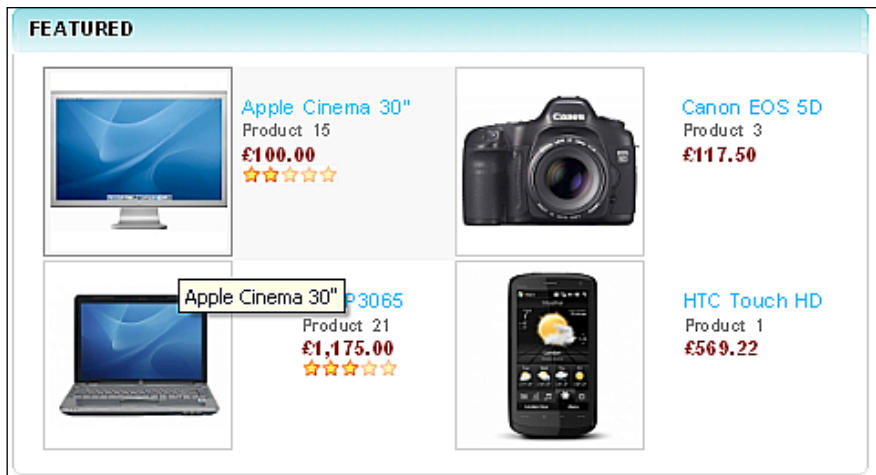
33. If we hover the product name now, it will change its text color; like this:



34. We may add styles to images, when we hover over them. You can add the following style block for it:

```
#featured-middle .product-image img:hover{ border: 1px solid #aFaFaF; }
```

35. And now go to the browser and hover over the product image to see the result:



## See also

There are other modules that can be placed at the middle section. We can apply these styles and make changes to those modules also.

## Applying styles to categories

Now, we will apply styles to our category block. We need to customize the current style and attributes. We place it on the left side of our store front.

### Getting started

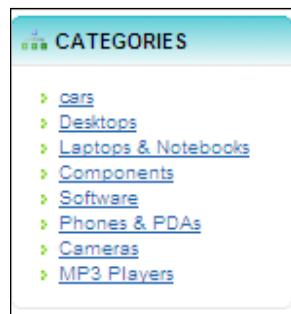
First, we will go to our admin panel. We use our stored credentials to get into the administration section. Then, go to **Extensions | Modules** to view category block. We will see the position, sort order, and status of the category block.

|          |      |         |   |                          |
|----------|------|---------|---|--------------------------|
| Category | Left | Enabled | 1 | <a href="#">[ Edit ]</a> |
|----------|------|---------|---|--------------------------|

Now, we will customize the style of the category block.

### How to do it...

1. First let's see the current style of the category block:



2. Open up the category tpl file; that is placed under the catalog\view\theme\shop\template\module.catalog\view\theme\shop\template\module:

```
<div class="box">
 <div class="top"><?php echo $heading_title; ?></div>
 <div id="category" class="middle"><?php echo $category; ?></div>
 <div class="bottom"> </div>
</div>
```

- This page is also divided into three sections: top, middle, and bottom. The top section contains the top header, the middle holds the category list, and the bottom layer contains the block bottom style.
- We will apply style to our category block. For inspection, we will use Firebug. Let's play with Firebug. Click on the **Firebug** icon, then click on the **Inspect** icon. Now, bring the mouse on the category block like this:



- When we click on this block, we will see the corresponding HTML tag on the console panel. The following image demonstrates this:

```
<div id="category">


```

- If you click on the `ul` tag, the style for this tag will be shown on the right side of the Firebug console panel:



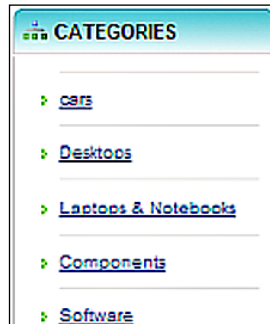
```
#category ul{
 list-style-image: url("../image/bullet_1.png");
 list-style-position: outside;
 list-style-type: disc;
 margin-bottom: 0;
 margin-left: 8px;
 margin-top: 0;
}
```

It is also showing the category style's line number in the `stylesheet.css` file. So, let's go to that line in our style file. We will apply the following style to the `li` tag:

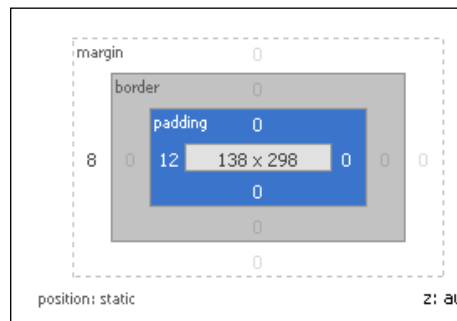
```
#category ul li{ padding: 10px 0px; border-top: 1px solid #eee;}
```



Here, we use 10px padding for top and bottom and 0px padding for the left and right side. And, there is a top border for `li`. When we see it in the browser, it will appear like the following:



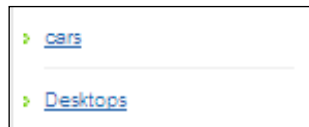
7. The list is slightly away from the left side. If we examine with Firebug, we find that there is a left margin and left padding:



We can check the adjustment by removing the left padding and margin. We make an adjustment in our category `ul` style like this:

```
padding-left: 0px;
```

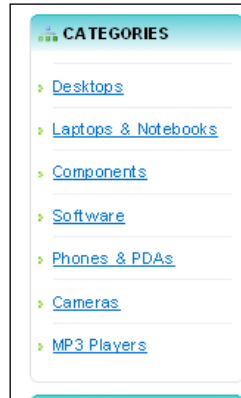
And it will be shown like this now:



8. You can apply some hover effect on the `li` tag. For this, we change the background color on mouse hover with the following style attribute:

```
#category ul li:hover { background-color: #f7f7f7;}
```

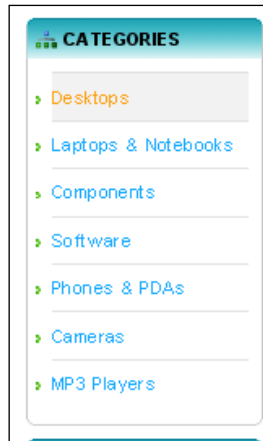
So, our category block will become like this:



9. We will change the font style and color of the category block. We change the text decoration and text color. You can use the following style attribute for it:

```
#category ul li a { text-decoration: none; color: #3FCDFD; }
```

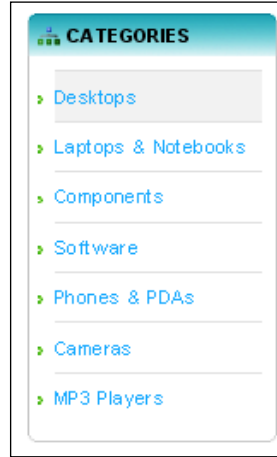
This will make our category block look like the following:



10. Now the text needs to change on mouse hover. We apply a different color to the text for hover effect. Use the following style for it:

```
#category ul li a:hover { color: #FFC62F; }
```

So, we have our category block like this:



Our complete style code block for category module stands like this:

```
/* ---- start of category block ----*/
#category ul { margin-top: 0px; margin-bottom: 0px; margin-left:
8px; padding-left: 0px; list-style: url('../image/bullet_
1.png');}
#category ul li{ padding: 10px 0px; border-top: 1px solid #eee;}
#category ul li:hover { background-color: #f7f7f7;}
#category ul li a { text-decoration: none; color: #3FCDFD; }
#category ul li a:hover { color: #FFC62F; }
/* ---- end of category block ----*/
```

11. We can change the top header image in the same way we did for the featured block. Or we can remove the image also, like this:

```
<div class="top"><?php echo $heading_title; ?></div>
```

And, to change the icons on the left of each category, you can make the following changes in the category ul style:

```
list-style: url('../image/name-of-the-icon.png');
```

So, our complete style code for this module block becomes this:

```
/* ---- start of middle- featured block ----*/
#featured-middle {padding: 0px 0px 0px 10px; list-style-type:
none; width: 100%; height: 260px;}
#featured-middle li {width: 50%; float:left; display: inline;}
#featured-middle li:hover { background-color: #f7f7f7;}
#featured-middle li a { text-decoration: none; color: #3FCDFD; }
#featured-middle li a:hover { color: #FFC62F; }
#featured-middle .product-image {float: left;}
#featured-middle .product-description {float:right; padding: 20px
30px 0px 0px;}
#featured-middle .product-image img{ border: 1px solid #ddd;
padding: 3px;}
#featured-middle .product-image img:hover{ border: 1px solid
#aFaFaF;}
/* ---- end of middle- featured block ----*/
```



# 5

## Header Structure

In this chapter, we will cover:

- ▶ Discussion of the existing header structure
- ▶ Making a wide header
- ▶ Logo positioning
- ▶ Modification of top navigation
- ▶ Styling menus
- ▶ Controlling the search bar
- ▶ Adjusting the language bar and currency units
- ▶ Header banner creation

### Introduction

We have seen module management and their layout modification in the previous chapter. Here, we are going to see how you can control the structure of the header of OpenCart. The header holds different parts.

This chapter will present to us the different parts of OpenCart header. We will customize each section and learn how we can control them. This chapter will be a step-by-step guide for header modification.

We will create the header banner for our store shop. We'll also make changes to top navigation, menus, search bar, language, and currency units. This chapter shows us how we can adjust the placement of our store logo.

## Default header structure

OpenCart header consists of several classes. These classes hold different sections of the header. It has **div1** to **div10** class. Each class controls different parts of the site. **div1** class has the logo of the site. **div3** maintains the top-right navigation. Menus are positioned with the **div4** class. **div5** has the search bar and **div6** holds the language bar, while **div7** controls the language and currency units. We will see how we can customize each class to make our header better.

## Altering the logo

**div1** class contains the logo of our store **shop**. In this recipe, we are going to see how we can adjust the position of the logo with the **div1** class. We will apply different style attributes to the class and also do some magic with GIMP.

## Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. For this, go to `catalog/view/theme/shop/stylesheet`.

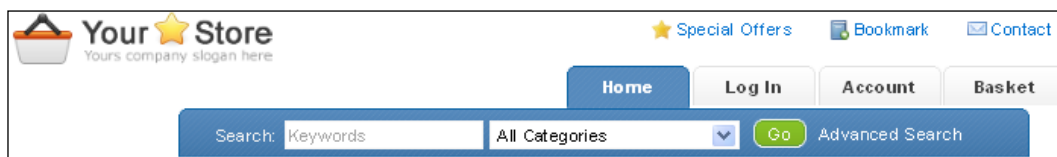
## How to do it

We will do the following steps for this recipe:

1. We make the height of the **div1** class to **200px**. You can make it according to your choice. For this, we need to add the following style to the stylesheet. There are no styles defined for the **div1** class in the default stylesheet:

```
#header .div1 {
height: 200px;
}
```

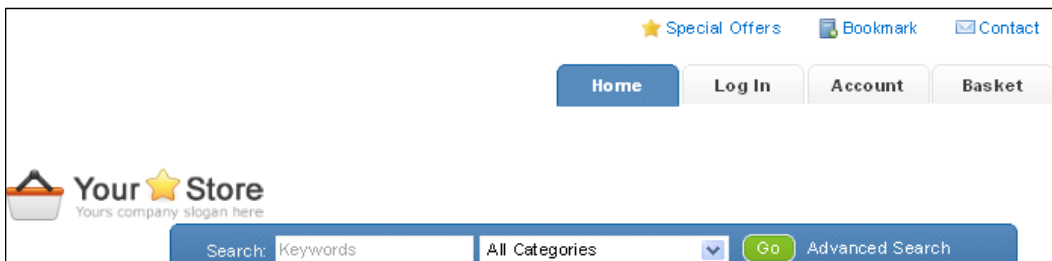
2. This will make the height of the header larger, like the following:



3. We need to change the **padding-top** amount to **128px**. So, the code style becomes like this:

```
#header .div2 {
 float: left;
 padding-top: 128px;
}
```

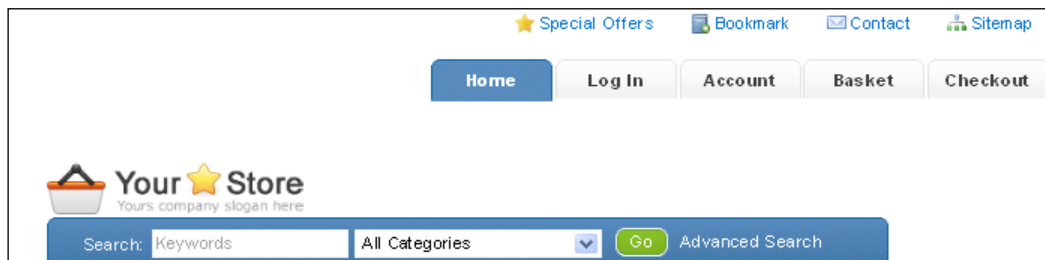
4. Adding this style brings the logo down a little:



5. And change the **padding-left** properties to **14%**. We use percent as our unit:

```
#header .div2 {
 float: left;
 padding-top: 128px;
 padding-left: 14%;
}
```

6. This makes the header like this:



## Styling top navigation

In this recipe, we will style the top right navigation of our store **shop**. We are going to create a background image for it. We will use **GIMP** for this purpose. Also, we apply some styling to the **div3** class. This class holds the top right navigational block.



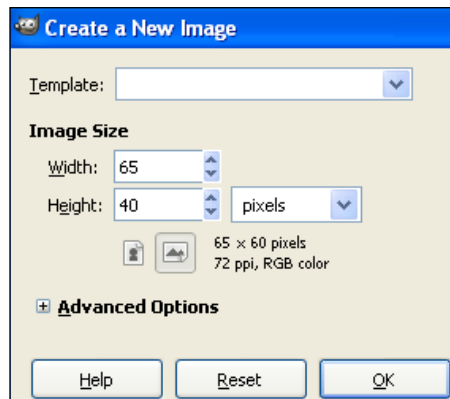
## Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to `catalog\view\theme\shop\stylesheet`.

## How to do it

Let's do the following steps:

1. As we have seen earlier, **div1** class consists of several other classes. We will change the properties of each sub class. Now, we will change the **div3** class styles of the header.
2. We will use a background image for this class. Let's create that image with GIMP. First, we will set the height and width of the image. We set it **65px** and **40px** respectively.



3. Select the **bucket fill** tool from the toolbox:



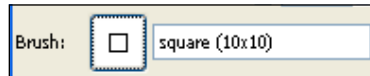
4. From the color panel, we select color **#d1d4d4**. But, you can choose your favorite color. Now, fill our image with the selected color. It will become like this:



5. We will apply some effects with brushes. So, select the **brush** icon:



6. There are many different types of brushes. Also, we can make our own. But for now, we will use **square (10x10)**:



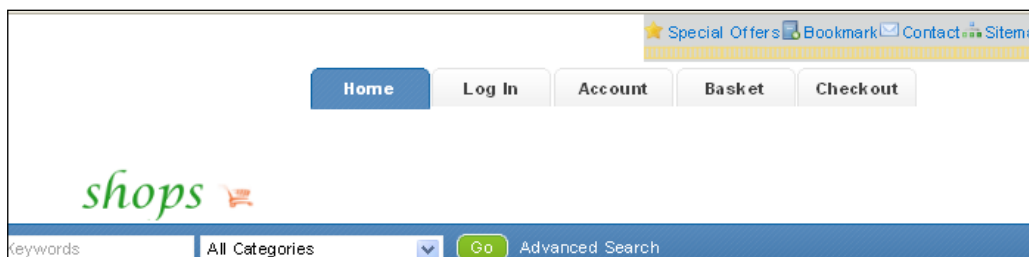
7. We are going to use the selected brush with a selected color. Click the color panel again. We will choose **#f6d666**. Our image icon becomes like this:



8. Now, we will use this image as the background image for the **div3** class. We'll add a background image property to the stylesheet. We named the image as **top-nav.png** and placed it under the `catalog\view\theme\sho\image`. So, the style code becomes the following for the **div3** class:

```
#header .div3 {
 float: right;
 padding-top: 7px;
 height: 38px;
 background: transparent url('../image/top-nav.png') repeat-x;
}
```

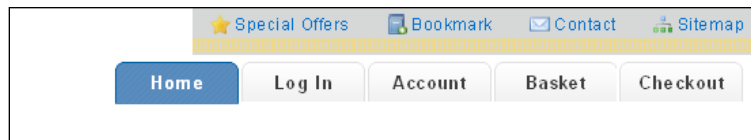
9. We repeat the image towards the x axis. In the browser, we see the changes of the **div3** class:



10. To adjust the position of **div3** class, we need to make some changes to the **margin-left** attribute—we remove that and use **margin** for it.

```
#header .div3 a {
 margin: 15px;
 padding: 1px 0px 2px 20px;
 background-repeat: no-repeat;
 background-position: left center;
}
```

11. The **div3** class will be shifted to left side. See the following image:



## Placing menus

Menus are an important part of a site. OpenCart has some default menus. We will style it in this recipe. **div4** class contains the navigational menus. Earlier, we saw top right navigational menus. Now, we will see main navigational menus.

### Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to `catalog\view\theme\shop\stylesheet`.

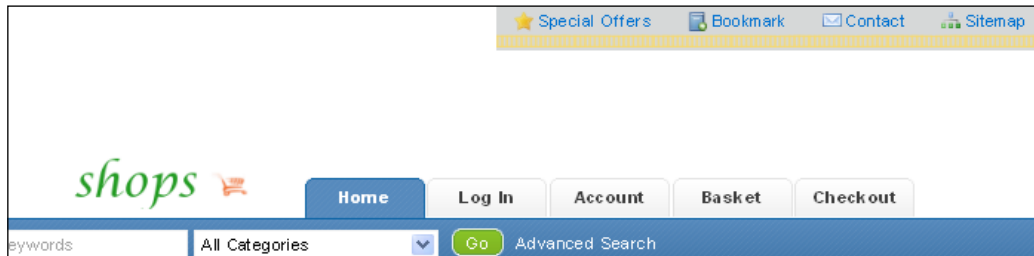
### How to do it

We will go with the following steps:

1. Now, we make style changes for the **div4** class. The **div4** class contains the navigational menus. We change the **padding-right** property to **14%**.

```
#header .div4 {
 width: 510px;
 clear: right;
 float: right;
 height: 32px;
 padding-right: 14%;
}
```

2. This will bring the menus down a little. You can find the changes like this:



3. We want to show our header a little bigger. So, we add a padding-top attribute to the **div4** class in the stylesheet. So, the style code for **div4** class becomes like this:

```
#header .div4 {
 width: 510px;
 clear: right;
 float: right;
 height: 32px;
 padding-right: 14%;
 padding-top: 100px;
}
```

4. And this makes our site look like the following image:



5. Now, the logo moves to the center of the header.

## Styling the search bar

This recipe will guide us through the styling of **div5** class. This class contains the search bar. At first, we adjust the width of the search bar. We use **percentage** unit for this. Later, we'll see how we can create an image for the search bar and replace it.

### Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to this: `catalog\view\theme\shop\stylesheet`.

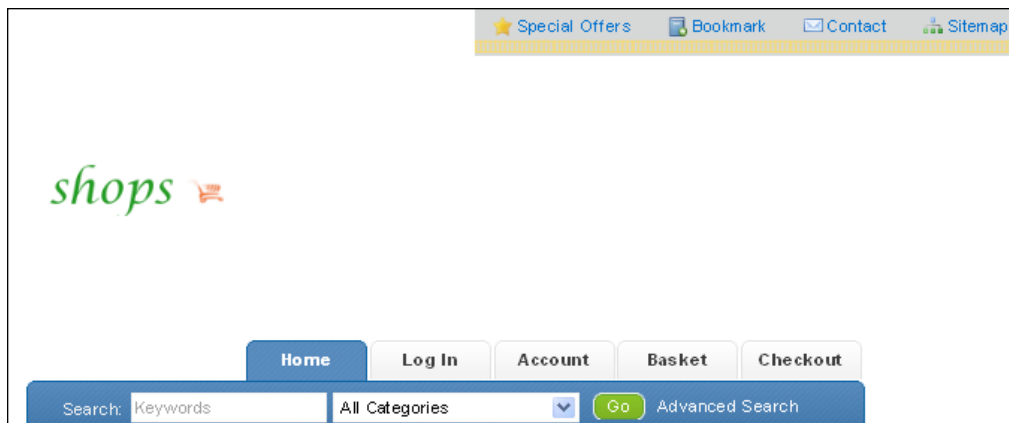
### How to do it

We will go with the following steps:

1. We now change the style for **div5** class. We set **0%** margin for top and bottom; **14%** margin for left and right.

```
#header .div5 {
 clear: both;
 margin : 0 14%;
}
```

2. Applying this will squeeze our navigational menu. See the following image:



3. **div5** class contains the search bar for our shop. We will change the color of this bar. Let's see how we can do this with **GIMP**.
4. The search bar is divided into three sections: **left**, **center**, and **right**.

5. For the left side, the default style code is as follows:

```
#header .div5 .left {
 background: url('../image/header_1_left.png') no-repeat;
 width: 5px;
 height: 40px;
 float: left;
}
```

6. To change the image, go to `catalog\view\theme\shop\image`. Open up **header\_1\_left.png** in GIMP. The image will be pretty narrow, so, we will **zoom in**. Click the **zoom** tool from the toolbox:



7. Then, click several times on the image, it will become larger:



8. We first change the color of the outside area. We select the outer boundary with **select by color** tool:



9. Now, click on the boundary area so that it will be selected. See the following image:



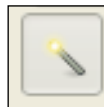
10. We pick the color that we want to fill with. Click on the color panel. We choose **#0599c4**. You can choose your favorite color.
11. We will fill the border with the **bucket fill** tool. So, select it from the toolbox:



12. And click on the selected boundary region to fill it with the selected color. Now, our left side image will look like this:



13. We will color the center region of our image. You need to select the **fuzzy select tool** from the toolbox:



14. To fill the region, again, select the Bucket Tool and click on the region. So, our left image is done. We will save it.



15. If we refresh our browser now, we will see our left image in action:



16. We will change the **center** and **right** image in the same way and save them. The style code block for **right** class of **div5** class is the following:

```
#header .div5 .right {
 background: url('../image/header_1_right.png') no-repeat;
 width: 5px;
 height: 40px;
 float: right;
}
```

17. Pick the image and edit it with GIMP. We will end up with the following image if you follow the above steps:



18. After saving the image, the search bar area will be like this:



19. You will find our new image on the left and right side of the search bar.
20. To change the center image, you need to open the image with GIMP. Get the location of the image from the corresponding style code:

```
#header .div5 .center {
 background: url('../image/header_1_center.png') repeat-x;
 height: 40px;
 margin-left: 5px;
 margin-right: 5px;
}
```



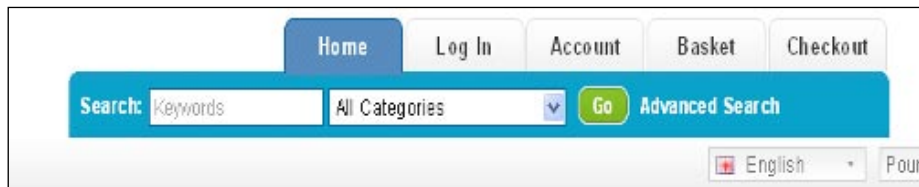
21. We need to change the color of the top, middle, and bottom section of the center image. The process is the same as the above two processes. After changing the top and bottom color, the image will become like this:



22. After changing the color in the middle region, the image will become like this:



23. If we go to the browser, then the site will look like this:



## Styling the language bar

There are language and payment units below the search bar. **div6** class controls the styles for the language bar. We will apply some styles in this recipe. Like the previous recipe, we will also create images for this class. This will be a step-by-step guide to create an image for **div6** class background color.

### Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to `catalog\view\theme\shop\stylesheet`.

## How to do it

See the following steps for this recipe:

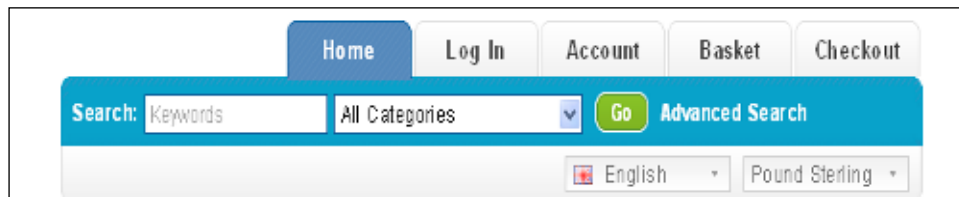
1. In the **div6** class, we have the **language** and **payment unit** options. We shorten **div5** class, so, we also need to make adjustment for the **div6** class. Currently, it has only **margin-bottom** attributes.

```
#header .div6 {
 clear: both;
 margin-bottom: 10px;
}
```

2. We will remove **margin-bottom**. First, we set **margin** to 0px 14% 10px. So, the code block becomes this:

```
#header .div6 {
 clear: both;
 margin : 0 14% 10px;
}
```

3. Then, the language bar becomes like this:



4. Like **div5** class, **div6** class also has three different parts: **left**, **right**, and **center**. We can find the location of the left image from the style code:

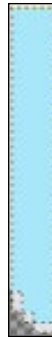
```
#header .div6 .left {
 background: url('../image/header_2_left.png') no-repeat;
 width: 5px;
 height: 32px;
 float: left;
}
```

5. We will open the image with GIMP. Let's change the image color to our needs.

6. As the image is too small to work with, we will enlarge the image with **zoom in**. Then, the image will become like the following:



7. As there will be a problem selecting the boundary area first, we will select the center region first. We will choose the **select by color** tool from the toolbox. Then, the area will be selected.
8. To fill the region, we select **#9fe4f8** from the color panel. Using the **Bucket Tool** from the toolbox, we will pour the color in the selected region. Then, our image will become like this:



9. Now, with the **select by color** tool of GIMP, select the boundary area and pour our chosen color into that area. So, our image stands like this:



10. When we see it in the browser, we found the change of our shop:



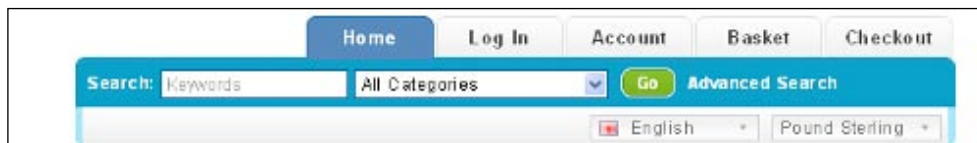
11. In the same way, we will change the **right** and **center** images also. First, we need to get the location of the right image. We can find it from style code block.

```
#header .div6 .right {
 background: url('../image/header_2_right.png') no-repeat;
 width: 5px;
 height: 32px;
 float: right;
}
```

12. By applying the above procedures, our right image becomes the following:



13. And, in the browser, the changes will appear as well:



14. Now, open up the center image. You can get the location of the image from the stylesheet as well:

```
#header .div6 .center {
 background: url('../image/header_2_center.png') repeat-x;
 height: 32px;
}
```

## Header Structure

---

```
margin-left: 5px;
margin-right: 5px;
padding-left: 5px;
padding-right: 5px;
}
```

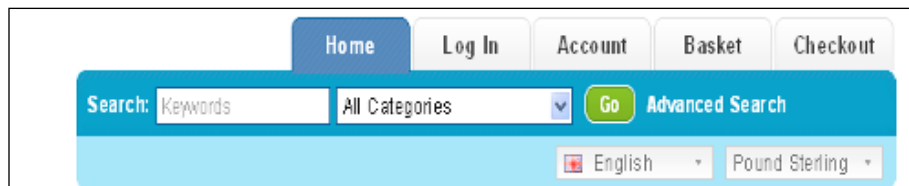
15. For the center image, we will change the top section first. Select the region with the **Select by color** tool from the toolbox. Using the Bucket Tool, we fill the area with our selected color **#9fe4f8**. And our image now looks like this:



16. After filling the top section, select the bottom with the **Select by color** tool and fill it with our color. So, the center image looks like this:



17. When we go to the browser, it will appear as the following:



18. **div7** class contains the language and payment units. You can place those units on the left side by changing the **float** property to the **left**. But we leave it as it is. We will apply a right padding to this class.

```
#header .div7 {
 float: right;
 padding: 2px 70px;
}
```

19. This will move the language and payment units to the left. You can see the change in the browser:



## Controlling search style

Earlier we changed the width and background image of the search bar, in this recipe, we will adjust the fonts and hover styling for the search units. There is a **search** ID to control these properties. We apply different changes to this ID. We also add extra attributes to HTML elements to apply additional styles.

### Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to `catalog\view\theme\shop\stylesheet`.

### How to do it

Followings are the steps needed in this recipe:

1. To change the search bar styling, we customize the **Search** ID. We change the font weight to **Normal**, give left padding to **25px** and change the color to **#F1F1F1**:

```
#search {
 color: #F1F1F1;
 font-weight: normal;
 padding-left: 25px;
 padding-top: 7px;
}
```

If we refresh our browser, we can easily find the changes:



2. In the **div8** class, which contains the search label, we apply 8px top:

```
#header .div8 {
 float: left;
 margin-top: 8px;
}
```

You will see that the search label gets lowered a little:



3. Now, we will add an extra style for the keyword input field: **#filter\_keyword**. See the following style code:

```
#search #filter_keyword {
 border: 1px solid #F1F1F1;
 padding: 2px;
}
```

4. Here, we apply a border to the keyword input field and a padding of **2px**. In the browser, you can easily view the changes:



5. Now, we apply style to the **category** list select box. We change the border color to **#f1f1f1**:

```
#search select {
 border: 1px solid #f1f1f1;
 padding: 1px;
}
```

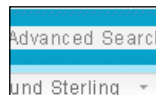
The select box for category becomes like this after applying the above styling:



- There is a link for advance search option. We will add a hovering style to the link. We have added color and background-color to hover effect. See the following code block:

```
#search a:hover {
 color: #7a7a7a;
 background-color: #f1f1f1;
}
```

So, when we hover over the advance search block, we can see our new style in action:



## Styling language and payment units

Earlier, we have changed the width and background image of the search bar. In this recipe, we will adjust the fonts and hover styling for the language and currency units. There is a **switcher** ID to control these properties. We apply different changes to this ID. We also add extra attributes to HTML elements to apply additional styles.

### Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to `catalog\view\theme\shop\stylesheet`.

### How to do it

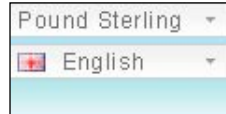
Followings are the steps to be followed in this recipe:

- There is a class named **switcher**. It holds the language and currency units. If we want to show the language and currency unit in a single column, we just remove the float property from the `switcher` class. The code will be like this for the **switcher** class:

```
.switcher {
 margin-top: 3px;
 margin-left: 10px;
}
```



Then, it will show this:



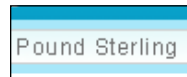
2. But we will not use this style for our **shop** site. So, we restore the float attribute. If you want to change the image of the language and currency unit, get the location of the image from the style code:

```
.switcher .selected {
 background: #FFFFFF url('../image/switcher.png') repeat-x;
}
```

3. Open up the image with GIMP and make the changes as you want. We apply **#e2f7f8** and choose the **gradient** option to apply on the image:



We view the new image on the browser:



4. We will change the background color to transparent. So, the code will be:

```
.switcher .selected {
 background: transparent url('../image/switcher.png') repeat-x;
}
```

5. For font color in the switcher class, we adjust the following code style:

```
.switcher a {
 text-decoration: none;
 display: block;
 color: #888;
}
```

6. And remove the color style from `.switcher .selected a` and `.switcher .option a`. You need to remove the following style:

```
.switcher .selected a {
 color: #666666;
}
```

And this too:

```
.switcher .option a {
 color: #000;
}
```

7. When you go to the browser, you will find this:



8. From the switcher class, we remove the border style. You can keep it if you want. We remove it from the following code section:

```
.switcher .selected a {
 border: 1px solid #CCCCCC;
}
```

Then, see the effect in the browser:



9. On hover, we change the background color to #c0eeff:

```
.switcher .selected a:hover {
 background: #c0eeff url('../image/arrow_down.png') 116px
 center no-repeat;
}
```

10. When we hover over the language or currency block, it appears as:



11. In the drop-down window, we change left and right border color to #f1f1f1, bottom border width to 7px, and change the background-color to #d5f6ff.

```
.switcher .option {
 position: absolute;
 z-index: 3;
 border-left: 1px solid #f1f1f1;
 border-right: 1px solid #f1f1f1;
 border-bottom: 7px solid #d5f6ff;
}
```

```
background-color: #d5f6ff;
display: none;
width: 131px;
}
```

Now, refresh our browser to see the change:



12. When we hover over our currency units, we change the background color to #c0eeff:

```
.switcher .option a:hover {
 background: #c0eeff;
}
```

We can view the change:



## Creating the header banner

Earlier, we changed the width and background image of the search bar. In this recipe, we will adjust the fonts and hover styling for the search units. There is a **search** ID to control these properties. We apply different changes to this ID. We also add extra attributes to the HTML elements to apply additional styles.

### Getting started

We need to open up our stylesheet, `stylesheet.css`, in our favourite editor. Go to `catalog\view\theme\shop\stylesheet`.

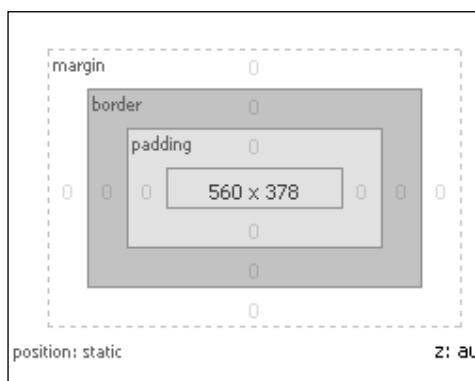
## How to do it

Followings are the steps needed in this recipe:

1. To create the header banner, first, we need to get the banner dimension. We will use **Firebug** to get the dimension:



2. Now, with the **inspector** selected, click the **header** div ID. We will get the dimension in the layout section of Firebug:



3. So, we will create a banner with height **378px** and width **96px**. We are using a smaller width and we will use the **repeat-x** property so that the image will cover all the header area.
4. Now, open up GIMP and create an image with our chosen dimension.
5. We select foreground and background colors in the color panel. For this recipe, we chose **#0d9492** and **#32b7c3** as foreground and background colors respectively. So, our color panel becomes like this:



6. We will select the **Gradient** tool from the tool box:



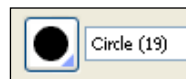
7. If we apply the gradient from top down, then the image will become like this:



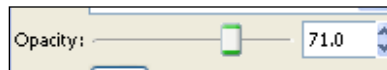
8. We are going to use different types of brushes. So, first select the brush from the tool box:



9. From brush tool options, we will choose **circle(19)**:



10. You can select your favorite brush. We set the opacity to **71**:



11. We choose **#c6a50c** for our brush. Now, apply the brush in the image. The banner becomes like this:



12. We change the opacity and scale to **60** and **1.50** and place the brush on the image. It produces this:



13. If we gradually decrease the opacity and scale, then the image becomes like the following:



14. We now change the color of our brush. We will make it **#ffffff**. And change the opacity to **100** and scale to **1.7**. Use the brush in our banner. So, it is now like this:



15. Again, we gradually decrease the scale and use the brush in our banner. So, it produces the following image:



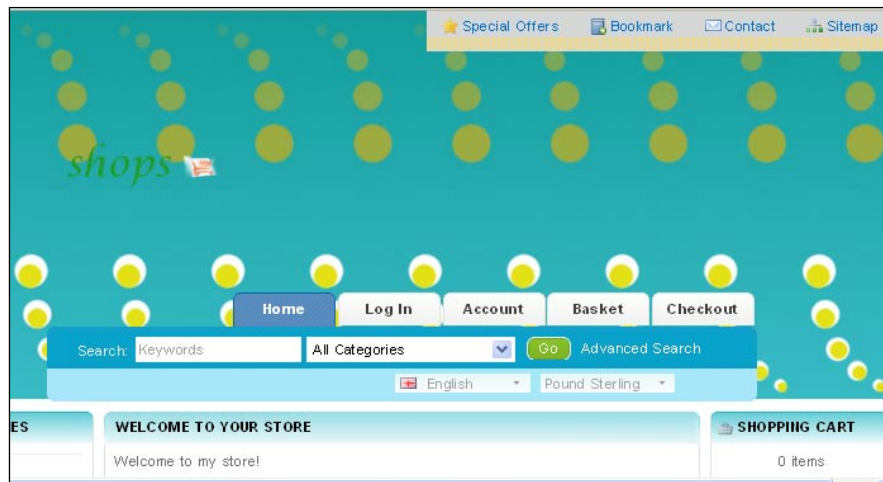
16. We change the brush color to **#e1df10** and apply it, gradually decreasing the scale on the image. We place the newly created brush circles on the white circles. So, our banner looks like this:



17. Place the image under `catalog\view\theme\shop\image`. We need to add some cascading styles for our header. The width needs to cover the whole header area. So, we set the width as **100%**:

```
#header {
 width: 100%;
 margin: auto;
 background: transparent url('../image/header-back.png') repeat-
x;
}
```

18. And set the margin as auto. We use the newly created banner as the background of header block. We repeat the image towards the x axis using **repeat-x** property. The background color will be transparent.
19. When we open our site in the browser after applying the above change, we see the header banner as the following:



# 6

## Dynamic Content

In this chapter, we will cover:

- ▶ Installing jQuery and jCarousel
- ▶ Displaying products with jCarousel
- ▶ Displaying products on jCarousel vertically
- ▶ Customizing the jQuery cycle plugin
- ▶ Modifying the side column with the jQuery cycle plugin
- ▶ Different transition effects of jQuery cycle plugin

### Introduction

Dynamic content makes sites more lively and attractive to users. We will focus on dynamic content in our store shop. We will use the jQuery JavaScript framework for this purpose. jQuery is easy to use and easy to learn. It has some features over normal JavaScript. We will not focus on those features of jQuery here, as it is out of the scope of this chapter. We will see how we can use jQuery with our OpenCart store step by step.

In the previous chapter, we have seen header creation and customization. Now, we move deeper into OpenCart theme creation. We will see how we can add dynamic contents in our site. We see the step by step addition of these dynamic contents in our site.

In this chapter, we will learn to use two popular jQuery plugins: JCarousel and jQuery **cycle**. We'll download the files and use it in a very easy way in this chapter.



## Installing jQuery and jCarousel

In this recipe, we will download jQuery and jCarousel and also install them into our store. By default, OpenCart comes with jQuery installed.

Carousels have become a popular means to display products. There are many different types of carousels. But, we will use jCarousel for this recipe. You can use your favourite one.

### Getting started

We need jQuery for our dynamic content throughout the site. jQuery is added by default with OpenCart. But here, we will also see how to download it and install it. We will download and install jQuery and jCarousel. First, we will go to the sites from where we can download them. If jQuery is not present in your site, you can download it from [http://docs.jquery.com/Downloading\\_jquery](http://docs.jquery.com/Downloading_jquery). We will download the latest jQuery JavaScript file.

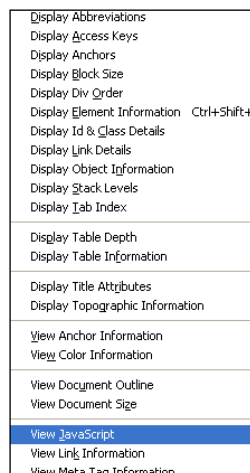
### How to do it

First, we need to download the required files to use jCarousel. We need the jQuery JavaScript file and jCarousel JavaScript and css files. To check whether our site is jQuery-enabled or not, we can use web developer Firefox addons. We installed this addon in Chapter 1 of this book.

Click on the **information** tab of Web Developer—the Firefox addon that we installed in Chapter 1.



It will display many sub-menus. Every sub-menu has its own importance. We select **View JavaScript**.



Then, a new tab will open containing all the JavaScripts for the web page in it. You can contract or expand the links. If our store already has jQuery, then we can find it in this page. Otherwise, it won't be in this page or it will show a **404** error.

```
ipt/jquery/jquery-1.3.2.min.js
ipt/jquery/thickbox/thickbox-
```

Since our store is jQuery-enabled by default, we don't need to install jQuery. We have shown the installation of jQuery so that you can have full control of the process. There are several releases available to download. After downloading jQuery, we will place it under the `catalog\view\Javascript\jQuery` folder.

Now, we will download jCarousel. We can download it from <http://sorgalla.com/projects/jCarousel/>.

Then, extract the compressed file. There will be many files in the extracted folder. Under the `lib` folder, we have an uncompressed and minified version of jCarousel. We create a folder named `jCarousel` under `catalog\view\javascript\jquery`. Then, in the `jCarousel` folder, create another folder named `js`. We will place any one of the two files under `catalog\view\javascript\jquery\jcarousel\js`.

And bring the `skins` folder under the `catalog\view\javascript\jquery\jcarousel`.

## See also

- Creating a logo

## Displaying products using jCarousel

We have installed jQuery and jCarousel in the previous recipe. Here, we will see how we can display products with jCarousel. In this recipe, we are going use jCarousel for the latest products, but you can use it for other modules as well.

## Getting started

First, we will open the `header.tpl` file under `catalog\view\theme\shop\template\common`. If we don't have jQuery in our site, then we need to add it here. See the following code block:

```
<script type="text/javascript" src="catalog/view/javascript/
jquery/jquery-1.3.2.min.js"></script>
```

## How to do it

We will follow the step below for jCarousel addition to our store:

1. First, we need to add the links for jCarousel JavaScript and css files to work with jCarousel. As we are using jCarousel for displaying the latest products, we place the links for those files in the `latest_home.tpl` under `catalog\view\theme\shop\template\module`. We add the following links:

```
<script type="text/javascript" src="catalog/view/javascript/
jquery/jcarousel/lib/jquery.jcarousel.js"></script>
```

```
<link rel="stylesheet" type="text/css" href="catalog/view/
javascript/jquery/jcarousel/skins/tango/skin.css" />
```

2. Now, we modify some HTML element structure to place the products in jCarousel. The current structure is `table` based. We will use `<ul>` `<li>` instead of the `table` structure. We remove the following tag:

```
<table class="list">
//...
</table>
```

3. And write the following code, here by the three dots, we mean other inner codes:

```
<ul id="latestcarousel" class="jCarousel-skin-tango">
//...

```

4. Here **latestcarousel** ID is our carousel container ID. There are two skins available for jCarousel, one is **tango** and the other is **ie7**. Here, we are using **tango**.

5. We also remove the **tr** tag:

```
<tr>
//...
</tr>
```

6. Now, remove the **td** tag and the following:

```
<td style="width: 25%;"><?php if (isset($products[$j])) { ?>
//...
<?php } ?></td>
```

7. And, we replace it with the following code:

```

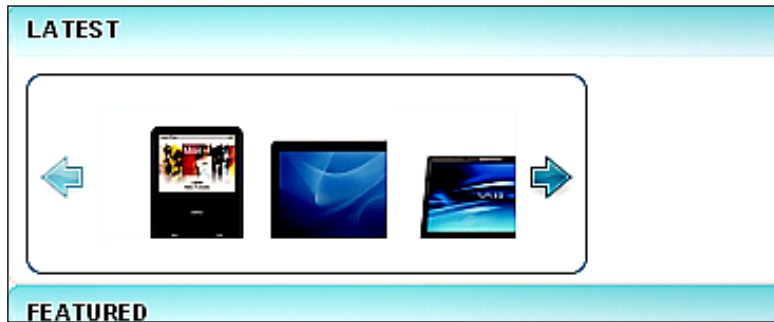
//...

```

8. Now, we will initialize jCarousel.

```
<script type="text/javascript">
jQuery(document).ready(function() {
 jQuery('#latestcarousel').jcarouseljcarousel();
});
</script>
```

If we see our changes in the browser, then we will find it as:



9. Here, we need to adjust the height and width of the carousel.
10. To change the width of the carousel, open the `skin.css` file under `catalog\iew\Javascript\jQuery\jCarousel\skins`. We are going to change the following code:

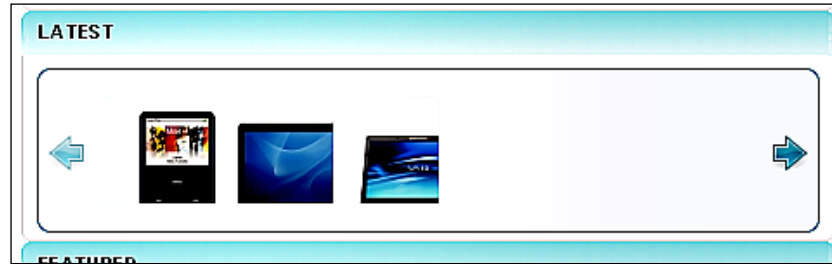
```
.jcarousel-skin-tango .jcarousel-container-horizontal {
 width: 245px;
 padding: 20px 40px;
}
```

To the following code:

```
.jcarousel-skin-tango .jcarousel-container-horizontal {
 width: auto;
 padding: 20px 40px;
}
```

Again, if we are going to use jCarousel in some other places as well, it is not smart to change the skin css. Instead we can override it in our theme-css for a specific region. For example, `#content .middle .jcarousel-skin-tango .jcarousel-container-horizontal`. Here, we have just shown you one instance of jCarousel usage.

We changed the width to **auto**. So, in the browser, the carousel will be like this:

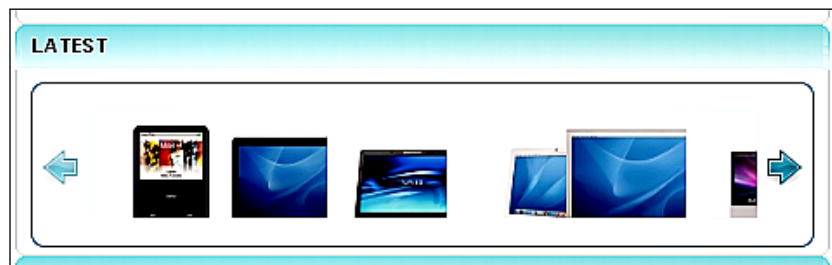


11. There is style for both vertical and horizontal carousel in the stylesheet. As we are now using the horizontal carousel, we adjust the styles for horizontal only. We will see the vertical carousel later in this chapter.

12. The area showing the images of the products is smaller compared to the carousel width. So, we will adjust the width. We set the width to **auto**.

```
.jcarousel-skin-tango .jcarousel-clip-horizontal {
 width: auto;
 height: 75px;
}
```

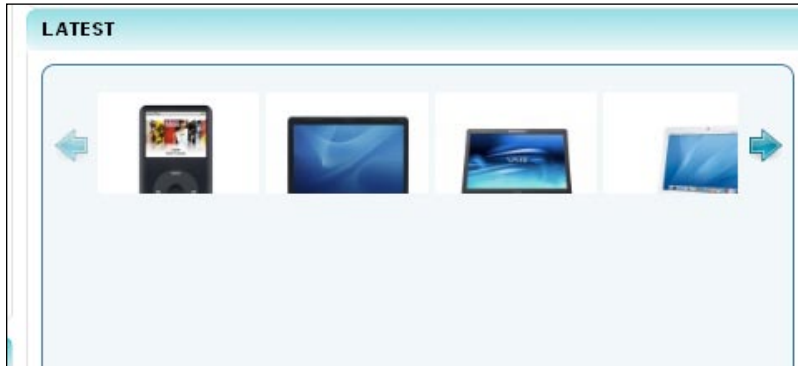
13. Now, the carousel is displaying products along the full width of the carousel. See the following image:



Our height for the carousel is small. Let's change the height of it. We change the height to **200px**:

```
.jcarousel-skin-tango .jcarousel-clip-horizontal {
 width: auto;
 height: 200px;
}
```

14. This makes our carousel look like the following:



15. To enlarge the height of the product image display area, we need to change the following code:

```
.jcarousel-skin-tango .jcarousel-item {
 width: 75px;
 height: 200px;
}
```

16. We change the height to **200px**. In the browser, now our carousel looks like this:



17. We need to adjust the margin property for our product images. We change the margin-right property to **50px**.

```
.jcarousel-skin-tango .jcarousel-item-horizontal {
 margin-left: 0;
 margin-right: 50px;
}
```

18. This makes spacing between the product images. You need to refresh the browser to view the changes:



19. We will set options for the carousel. There are several options. You can see the available options in this link: <http://sorgalla.com/projects/jcarousel/>.
20. We used the following options and placed it in the latest\_home.tpl, the page in which we are showing jCarousel:

```
<script type="text/javascript">
jQuery(document).ready(function() {
 jQuery('#latestcarousel').jcarousel({
 scroll: 1,
 visible: 3,
 auto: 3,
 rtl: true,
 wrap: 'circular'
 });
});
</script>
```

21. Let's discuss the options that we have used in our carousel:

- ☐ **Scroll:** With this option, we set how many images will be scrolled by a single rotation. We set it to **1**.
- ☐ **Visible:** This option tells how many images will be visible in the carousel. We set the visible product images to **3**.
- ☐ **Auto:** We set the number of seconds after which the carousel will be automatically rotated. We set it to **3**.
- ☐ **Rtl:** This option means right to left scroll. By default, the carousel scrolls from left to right.
- ☐ **Wrap:** If we set the wrap option to **Circular**, then the carousel will rotate as circular.

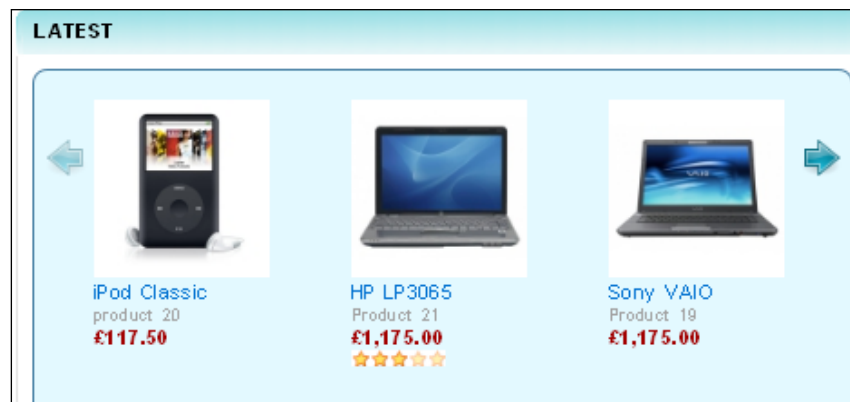
Now, let's see how our carousel looks:



22. If you want to change the colors of our carousel, then you need to adjust some styles. To change the background color of the carousel, you need to apply the following style:

```
.jcarousel-skin-tango .jcarousel-container {
 -moz-border-radius: 10px;
 background: #94f8f7;
 border: 1px solid #346F97;
}
```

23. This changes the background color to a shade of green:



24. And, change the border color to this:

```
border: 1px solid #abe9fd;
```



25. Then, navigational previous and next arrows are placed a little high in the carousel. We set the top property to **103px**.

```
.jcarousel-skin-tango .jcarousel-next-horizontal {
 position: absolute;
 top: 103px;
 right: 5px;
 width: 32px;
 height: 32px;
 cursor: pointer;
 background: transparent url(next-horizontal.png) no-repeat 0
0;
}
```

26. Also, for the previous button, the style changes to:

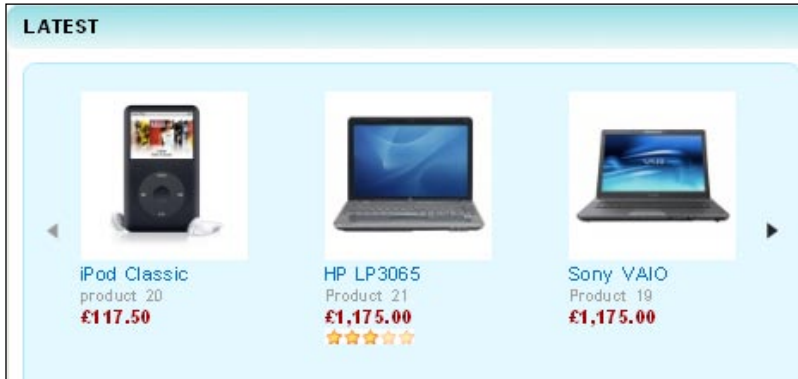
```
.jcarousel-skin-tango .jcarousel-prev-horizontal {
 position: absolute;
 top: 103px;
 left: 5px;
 width: 32px;
 height: 32px;
 cursor: pointer;
 background: transparent url(prev-horizontal.png) no-repeat 0
0;
}
```

27. The previous and next button position changes to this:



28. We will change the buttons now. You create new buttons or search the Internet for previous and next buttons. We place the buttons under `catalog\view\javascript\jquery\jcarousel\skins\tango`. You need to rename your buttons as `prev-horizontal.png` and `next-horizontal.png`.

29. Now, in the browser, our latest product carousel appears as the following:



### How it works...

## Displaying products vertically

We have seen the horizontal carousel so far. In this recipe, we will see how we can use the vertical carousel. We make the carousel horizontal for the home module of the latest products. Because on the left and right column, we have less space, we can use the vertical carousels there. It will utilize our space.

### Getting started

Open the `latest.tpl` file in our favourite editor under `catalog\view\theme\shop\template\module`.

### How to do it

To add the vertical carousel, we will follow these steps:

1. First, we will add the required stylesheet and JavaScript files. As this is a new recipe, we are adding jCarousel again:

```
<script type="text/javascript" src="catalog/view/javascript/
jQuery/jcarousel/lib/jquery.jcarousel.js"></script>
<link rel="stylesheet" type="text/css" href="catalog/view/
javascript/jquery/jcarousel/skins/tango/skin.css" />
```

2. You need to remove the table structure and replace it with ul li. See the following code block:

```
<table cellpadding="2" cellspacing="0" style="width: 100%;">
//...
</table>
```

3. Remove it and replace it with:

```
<ul id="latestcarousel" class="jcarousel-skin-tango">
//...

```

4. Also, we will remove the following code:

```
<tr>
 <td valign="top" width="1">
//...
 </td>
</tr>
```

5. We replace it with:

```

//...

```

6. We need to initialize the jCarousel with the following code:

```
<script type="text/javascript">
 jQuery(document).ready(function() {
 jQuery('#latestcarousel').jcarousel({
 scroll: 1,
 vertical: true,
 visible: 3,
 auto: 1,
 rtl: true,
 wrap: 'circular'
 });
 });
</script>
```

7. Here, we use the previous options as described earlier. Additionally, we use the **vertical** option. By setting this to true, we convert our carousel from **horizontal** to **vertical**.

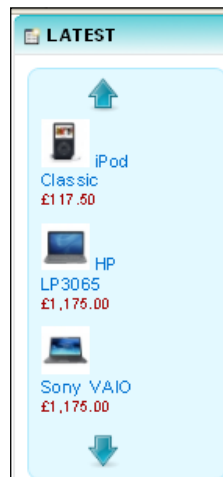
8. We will see our carousel in action in the browser:



9. Now, we will change the width of our carousel. For the vertical carousel, we need to set the width differently. We will change the width to auto in the following code:

```
.jcarousel-skin-tango .jcarousel-container-vertical {
 width: auto;
 height: 245px;
 padding: 40px 20px;
}
```

10. This makes our carousel like this:



11. The product images are on the left side. We increase the width of the carousel. So, the images will be placed in the center:

```
.jcarousel-skin-tango .jcarousel-clip-vertical {
 width: 135px;
 height: 245px;
}
```

12. This makes our carousel look like this:



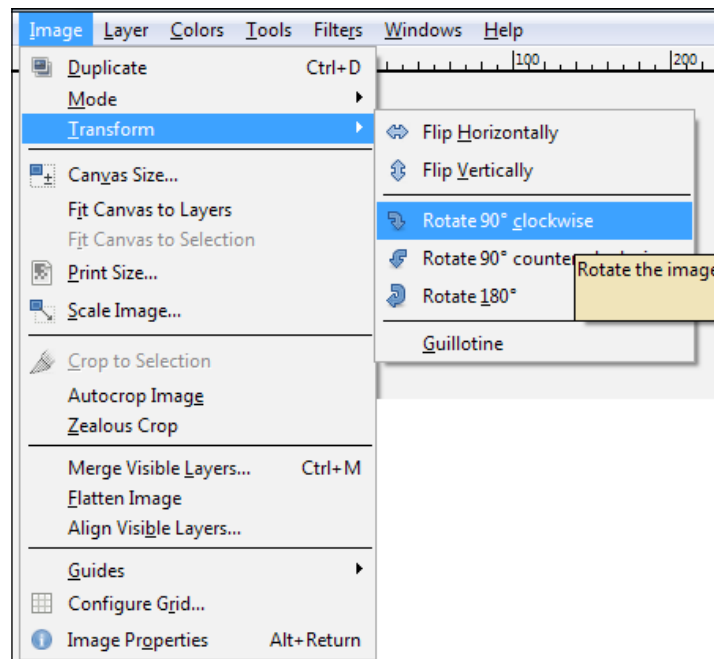
13. Our navigational button is slightly left justified. We will place it at the middle. We change the left attributes to **63px**.

```
.jcarousel-skin-tango .jcarousel-next-vertical {
 position: absolute;
 bottom: 5px;
 left: 63px;
 width: 32px;
 height: 32px;
 cursor: pointer;
 background: transparent url(next-vertical.png) no-repeat 0 0;
}
```

14. If we refresh our browser, we see the changes:



15. We will change the previously downloaded horizontal buttons to vertical. Open up the image with GIMP and create the duplicate image. Right click on the image. Then, go to **Image | Transform | Rotate 90° clockwise**. Save the image as `next-vertical.png`.



16. Again, open up our downloaded horizontal buttons and create the duplicate image. Right click on the image. Then, go to **Image | Transform | Rotate 90 Anti-clock-wise**. And, save the image as **prev-vertical.png**:



17. We are going to show the product image on the left side and the product description on the right side. We make the following code adjustment:

```
<div style="float: left;"><a href="<?php echo str_
replace('&', '&', $product['href']); ?>">" /></
a></div>

 <div style="float: right;"><a href="<?php echo str_
replace('&', '&', $product['href']); ?>"><?php echo
$product['name']; ?>
//...
<?php } ?></div>
```

18. As our product display width is small, we need to widen the area. We change the following code:

```
.jcarousel-skin-tango .jcarousel-item {
 width: 125px;
 height: 200px;
}
```

19. Then, our vertical carousel becomes like this:



20. We add border and padding top property to our carousel. We add the following code for this styling:

```
.jcarousel-skin-tango .jcarousel-item-vertical {
 margin-bottom: 10px;
 border: 1px solid #ddd;
 padding-top: 10px;
}
```

21. We also add a hover effect to our border:

```
.jcarousel-skin-tango .jcarousel-item-vertical:hover{
 border: 1px solid #aaa;
}
```

22. When we go to the browser, we find our changes:





## Customizing the cycle plugin

We have placed jCarousel to display our products. In this recipe, we use the jQuery cycle plugin with our products. We first download the jQuery cycle plugin.

### Getting started

We need jQuery enabled for jQuery cycle to work with our store **shop**. We have already shown how to install jQuery in an earlier recipe.

### How to do it

Go through the following steps to customize the jQuery cycle plugin:

1. First, download the jQuery cycle JavaScript files from <http://jquery.malsup.com/cycle/download.html>. We extract the downloaded compressed file. We will use the `jquery.cycle.min.js`. We copy this file to `catalog\view\javascript\jquery`.
2. We need to add jQuery and the jQuery cycle JavaScript file into our file. For this recipe, we will add this plugin for the latest products in the home section. So, we add the following code in `latest_home.tpl` as we are not using the jQuery cycle plugin throughout the site:

```
<script type="text/Javascript" src="catalog/view/javascript/jquery/jquery.cycle.min.js"></script>
```

3. Then, we will modify the existing **table**-based structure to **div**-based. We remove the following code:

```
<table class="list">
//...
</table>
```

4. And in place of that we write the following:

```
<div class="slideshow">
//...
</div>
```

5. We also again remove the **tr** tag:

```
<tr>
//...
</tr>
```

6. And replace the **td** HTML tag with the following **div** tag:

```
<td style="width: 25%;">
//...
</td>
```

7. The required **div** tag is:

```
<div class="slideshow-image-container">
//...
</div>
```

8. We will initialize the jQuery cycle plugin with the following code:

```
<script type="text/Javascript">
$(document).ready(function() {
 $('.slideshow').cycle({
 fx: 'fade'
 });
});
</script>
```

9. Now, go to the browser and refresh to see the effect of our change:



10. We center the product image. So, we add the following `margin-left` property to our image in the `stylesheets.css` file:

```
.slideshow .slideshow-image-container {
margin-left: 200px;
}
```

11. Then, the image container moves to the center of the latest product area. See the following image:



12. We need to do some styling to our product images. We will have a thick border around our image. So, we add these styling attributes:

```
.slideshow .slideshow-image-container {
 margin-left: 200px;
 border: 10px solid #ddd;
}
```

13. This creates a border around the product image like the following:



14. The product image and the descriptions are all left aligned. So, we make it centered by adding the following style tag:

```
.slideshow .slideshow-image-container {
 margin-left: 200px;
 text-align: center;
 padding: 10px;
 border: 10px solid #ddd;
}
```

15. Now, our jQuery cycle looks like this:



### There's more...

You can also see the Accordion jQuery plugin. It is also a very popular jQuery plugin. You can add and initialize it in almost the same way. You can read the documentation of the plugin at <http://plugins.jquery.com/project/accordion>.

## Modifying the side column with the cycle plugin

We modified the latest product block in the center column with the jQuery cycle plugin. Now, if we want to show our products on the side columns with the jQuery cycle, then this recipe will guide us to our destination.

### Getting started

We need jQuery enabled for jQuery cycle to work with our store **shop**. We have already shown how to install jQuery in an earlier recipe: *Installing jQuery and jCarousel*.

### How to do it

Go through the following steps to customize the jQuery cycle plugin:

1. To use the plugin, first, we need to add the required file links to the `latest.tpl` file as we are using the jQuery cycle for latest products. We add the following line in our `latest.tpl` file:
 

```
<script type="text/Javascript" src="catalog/view/javascript/jquery/jquery.cycle.all.min.js"></script>
```

2. Then, like our previous recipes, we will remove the `table`-based style and instead, use `div`-based styles:

```
<table cellpadding="2" cellspacing="0" style="width: 100%;">
//...
<tr>
//...
</tr>
//..
</table>
```

3. And, we write the following tag in place of the `table` element:

```
<div class="slideshow">
//...
</div>
```

4. We also `li` tag with a `div` element. See the following code:

```
//remove this tags

//...

//replace with the following element
<div class="slideshow-image-container">
//...
</div>
```

5. Now, we initialize the cycle plugin with the code below:

```
<script type="text/JavaScript">
$(document).ready(function() {
 $('.slideshow').cycle({
 fx: 'fade'

 });
});
</script>
```

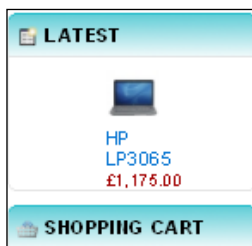
6. If we go to the browser, then we can see the following changes:



7. We will add some styling to the above display. The products are displaying on the left side of the side column. To make it centered, we add **margin** to the style:

```
.slideshow .slideshow-image-container { margin-left: 60px; }
```

8. Our right column changes like this:



9. We add a border to our image. We do the following styling addition:

```
.slideshow .slideshow-image-container {
margin-left: 60px;
border: 5px solid #ddd;
}
```

10. When we go to the browser, we find the following state of our right side bar:



11. We need to add some padding and make our text aligned to the center. So, we also add the following styles:

```
.slideshow .slideshow-image-container {
margin-left: 60px;
border: 5px solid #ddd;
padding: 5px;
text-align: center;
}
```

12. We refresh our browser and see our changes in action:



### There is more

You can also add this plugin on the left side of our OpenCart store. Just change the position of the module from the module section of the admin panel.

## Different effects with the cycle plugin

We have installed the jQuery cycle with our store **shop**. jQuery cycle comes up with many different options and effects. We can bring attractive changes to our product images with these options and effects.

### Getting started

We need jQuery enabled for the jQuery cycle to work with our store **shop**. We have already shown how to install jQuery in an earlier recipe.

### How to do it

Following is the list of available jQuery cycle options.

Let's discuss each option:

- **fx:** The name of the transition effect. We can use comma-separated names, e.g.: **fade, scrollUp, shuffle:**

```
fx: 'fade',
```

- **timeout:** We set the time between slide transitions. It is in milliseconds and 0 to disable auto advance:

```
timeout: 4000,
```

- ▶ **timeoutFn:** This is the callback for determining per-slide timeout value: `function(currSlideElement, nextSlideElement, options, forwardFlag):`  
`timeoutFn: null,`
- ▶ **Continuous:** We can set this to true to start the next transition immediately after the current one completes:  
`continuous: 0,`
- ▶ **Speed:** This is the speed of the transition. This can be any valid **fx** speed value:  
`speed: 1000,`
- ▶ **speedIn:** This is the speed of the **in** transition:  
`speedIn: null,`
- ▶ **speedOut:** This is the speed of the **out** transition:  
`speedOut: null,`
- ▶ **Next:** We use this as the selector for the element to use as a click trigger for the next slide:  
`next: null,`
- ▶ **Prev:** We use this as the selector for the element to use as click trigger for the previous slide:  
`prev: null,`
- ▶ **prevNextClick:** Callback function for prev/next clicks: `function(isNext, zeroBasedSlideIndex, slideElement)`  
`prevNextClick: null,`
- ▶ **prevNextEvent:** This is the event that drives the manual transition to the previous or next slide:  
`prevNextEvent: 'click.cycle',`
- ▶ **Pager:** This is the selector for element to use as pager container:  
`pager: null,`
- ▶ **pagerClick:** We can set callback function `fn` for pager clicks: `function(zeroBasedSlideIndex, slideElement).`  
`pagerClick: null,`
- ▶ **pagerEvent:** The name of the event that drives the pager navigation:  
`pagerEvent: 'click.cycle',`



- ▶ **allowPagerClickBubble:** This allows or prevents click event on pager anchors from bubbling:  
`allowPagerClickBubble: false,`
- ▶ **pagerAnchorBuilder:** This is the callback function for building anchor links:  
`function(index, DOMelement).`  
`pagerAnchorBuilder: null,`
- ▶ **before:** The transition callback (scope set to element to be shown):  
`function(currSlideElement, nextSlideElement, options, forwardFlag).`  
`before: null,`
- ▶ **after:** This is the transition callback (scope set to element that was shown):  
`function(currSlideElement, nextSlideElement, options, forwardFlag).`  
`after: null,`
- ▶ **end:** This callback is invoked when the slideshow terminates (use with autostop or nowrap options): `function(options).`  
`end: null,`
- ▶ **easing:** This is the easing method for both in and out transitions:  
`easing: null,`
- ▶ **easeIn:** Easing for **in** transition:  
`easeIn: null,`
- ▶ **easeOut:** Easing for **out** transition:  
`easeOut: null,`
- ▶ **shuffle:** We set co-ordinates for shuffle animation, e.g.: {top: 15, left: 200}.  
`shuffle: null,`
- ▶ **animIn:** This property defines how the slide animates in:  
`animIn: null,`
- ▶ **animOut:** This property defines how the slide animates out:  
`animOut: null,`
- ▶ **cssBefore:** This property defines the initial state of the slide before transitioning in:  
`cssBefore: null,`
- ▶ **cssAfter:** This property defines the state of the slide after transitioning out:  
`cssAfter: null,`

- ▶ **fxFn**: This function is used to control the transition: `function(currSlideElement, nextSlideElement, options, afterCallback, forwardFlag)`.  
`fxFn: null,`
- ▶ **height**: We set the container height:  
`height: 'auto',`
- ▶ **startingSlide**: Zero-based index of the first slide to be displayed:  
`startingSlide: 0,`
- ▶ **sync**: We set true if in/out transitions should occur simultaneously:  
`sync: 1,`
- ▶ **random**: We set true for random, false for sequence (not applicable to shuffle fx):  
`random: 0,`
- ▶ **fit**: We force slides to fit container:  
`fit: 0,`
- ▶ **containerResize**: We can resize the container to fit the largest slide:  
`containerResize: 1,`
- ▶ **pause**: We set true to enable pause on hover:  
`pause: 0,`
- ▶ **pauseOnPagerHover**: We set true to pause when hovering over the pager link:  
`pauseOnPagerHover: 0,`
- ▶ **autostop**: We set true to end slideshow after X transitions (where X == slide count):  
`autostop: 0,`
- ▶ **autostopCount**: This is the number of transitions. This is optionally used with autostop to define X.  
`autostopCount: 0,`
- ▶ **delay**: The additional delay (in ms) for first transition. This can be negative.  
`delay: 0,`
- ▶ **slideExpr**: This expression is for selecting slides; if something other than all children is required:  
`slideExpr: null,`
- ▶ **cleartype**: We set true if clearType corrections should be applied for IE.  
`cleartype: !$.support.opacity,`

- ▶ **cleartypeNoBg**: We set true to disable extra cleartype fixing; leave false to force background color setting on slides:  
`cleartypeNoBg: false,`
- ▶ **nowrap**: We can set true to prevent the slideshow from wrapping:  
`nowrap: 0,`
- ▶ **fastOnEvent**: This forces fast transitions when triggered manually via pager or prev/next; value is time in milliseconds:  
`fastOnEvent: 0,`
- ▶ **randomizeEffects**: This is valid when multiple effects are used; true to make the effect sequence random:  
`randomizeEffects: 1,`
- ▶ **Rev**: This causes animations to transition in reverse:  
`rev: 0,`
- ▶ **manualTrump**: This causes manual transition to stop an active transition instead of being ignored:  
`manualTrump: true,`
- ▶ **requeueTimeout**: We set the delay time for requeue. It is in milliseconds.  
`requeueTimeout: 250,`
- ▶ **activePagerClass**: This is the class name used for the active pager link:  
`activePagerClass: 'activeSlide',`
- ▶ **updateActivePagerLink**: We set a callback function fn. It is invoked to update the active pager link (adds/removes activePagerClass style):  
`updateActivePagerLink: null`

Now, we will see addition of some effects with our store **shop**.

1. First, let's see the **shuffle** effect in our shop. In the jQuery cycle initialization code block, we add the following code chunk:

```
<script type="text/Javascript">
$(document).ready(function() {
 $('.slideshow').cycle({
 fx: 'shuffle',
 delay: 20
 });
});
</script>
```

2. In the browser, we can find the following changes:



3. Now, we will see the **zoom** effect for the cycle plugin. We need to add the following code block in our initialization:

```
<script type="text/Javascript">
$(document).ready(function() {
 $('.slideshow').cycle({
 fx: 'zoom',
 sync: false,
 delay: 20
 });
});
</script>
```

4. If we go to the browser, then we can find **zoom** in action:



5. Again, we see the **toss** transition effect with our products. We need to add the following codes:

```
<script type="text/Javascript">
$(document).ready(function() {
 $('.slideshow').cycle({
 fx: 'toss',

```

```
 sync: true,
 speed: 2000,
 delay: 20
 });
});
</script>
```

6. We see the following effect in our store:



7. Now, let's see the effects of **growY** with our latest products. We add the following codes in our latest.tpl file:

```
<script type="text/Javascript">
$(document).ready(function() {
 $('.slideshow').cycle({
 fx: 'growY',
 sync: true,
 speed: 2000,
 delay: 20
 });
});
</script>
```

8. And in the browser, we see the changes:



## There's more...

We can also set different transition effects. We are showing it here, so you can use your favourite options for the cycle. The following is a list of transition effects in our store **shop**:

- ▶ blindX
- ▶ blindY
- ▶ blindZ
- ▶ cover
- ▶ curtainX
- ▶ curtainY
- ▶ fade
- ▶ fadeZoom
- ▶ growX
- ▶ growY
- ▶ none
- ▶ scrollUp
- ▶ scrollDown
- ▶ scrollLeft
- ▶ scrollRight
- ▶ scrollHorz
- ▶ scrollVert
- ▶ shuffle
- ▶ slideX
- ▶ slideY
- ▶ toss
- ▶ turnUp
- ▶ turnDown
- ▶ turnLeft
- ▶ turnRight
- ▶ uncover
- ▶ wipe
- ▶ zoom

You can use any of the above in the jQuery cycle. In this recipe, we will see some transition effects in action.

You can also use the other available options of the jQuery cycle in our OpenCart store. You can add them in the above mentioned way.

## **See also**

- [Installing jQuery and jCarousel](#)

# 7

## Customizing Menus

In this chapter we'll cover:

- ▶ Displaying the horizontal slider menu
- ▶ Image creation for the horizontal slider menu
- ▶ Integration of the fisheye menu
- ▶ Creating images and adjustment of styles for the fisheye menu
- ▶ Addition of total background menu
- ▶ Modification of social networking menus

### Introduction

So far, we have done a lot with different sections of OpenCart. Here, our primary concern will be decorating our store with modern menus. We are going to play with different types of menus in this chapter.

In the previous chapter, we saw how to customize our store with dynamic contents. This chapter will guide us through different types of menus. So, let's start decorating our store shop with menus.

### Displaying horizontal sliding menus

In this recipe, we will add a long wide menu into our store shop.

#### Getting started

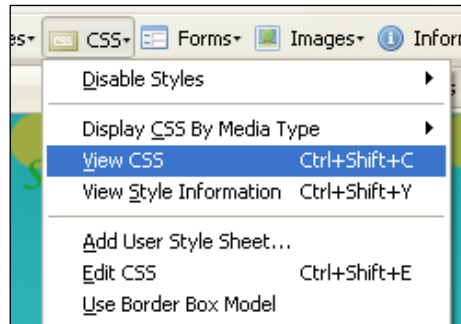
We will add a menu like the one shown in this link: <http://www.cssplay.co.uk/menus/css3-image-slide.html>



## How to do it

We will follow these steps to add the horizontal sliding menus into our site:

1. We will use the Web developer tool to get the cascading style of horizontal menu.
2. To get the style code, first open up <http://www.cssplay.co.uk/menus/css3-image-slide.html>.
3. Now, we will get the css style code from `css\view css`.



4. It will open up a new window with all the style code for the site. From there, we will find the inline style code for the site:

**<http://www.cssplay.co.uk/menus/css3-image-slide.html>**

**Embedded Styles from <http://www.cssplay.co.uk/menus/css3-image-slide.html>**

```
@import url(http://www.google.com/cse/api/branding.css);

.slideOuter {width:498px; height:200px; overflow:hidden; margin:0 auto;}
.slideOuter .slide {padding:0; margin:0; list-style:none; width:800px; height:200px; overf
.slideOuter .slide li {float:left;}
.slideOuter .slide li a {display:block; float:left; width:98px; border-right:2px solid #ff
-webkit-transition: 0.5s;
-moz-transition: 0.5s;
-o-transition: 0.5s;
transition: 0.5s;
}
.slideOuter .slide a#p5 {width:318px;}
.slideOuter .slide.current a {width:43px;}
.slideOuter .slide.current li a.current {width:318px;}

.slideOuter .slide li a img {border:0;}

.slideOuter .slide:hover a#p1,
.slideOuter .slide:hover a#p2,
.slideOuter .slide:hover a#p3,
.slideOuter .slide:hover a#p4 {width:43px;}

.slideOuter .slide a#p1:hover,
```

5. So, we will get the style code for `slideOuter` class:

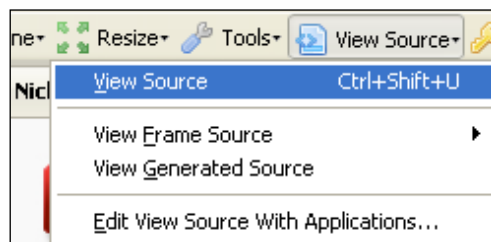
```
.slideOuter {width:498px; height:200px; overflow:hidden;
margin:0 auto;}
.slideOuter .slide {padding:0; margin:0; list-style:none;
width:800px; height:200px; overflow:hidden;}
.slideOuter .slide li {float:left;}
.slideOuter .slide li a {display:block; float:left; width:98px;
border-right:2px solid #fff; height:200px; overflow:hidden;
-webkit-transition: 0.5s;
-moz-transition: 0.5s;
-o-transition: 0.5s;
transition: 0.5s;
}
.slideOuter .slide a#p5 {width:318px;}
.slideOuter .slide.current a {width:43px;}
.slideOuter .slide.current li a.current {width:318px;}

.slideOuter .slide li a img {border:0;}

.slideOuter .slide:hover a#p1,
.slideOuter .slide:hover a#p2,
.slideOuter .slide:hover a#p3,
.slideOuter .slide:hover a#p4 {width:43px;}

.slideOuter .slide a#p1:hover,
.slideOuter .slide a#p2:hover,
.slideOuter .slide a#p3:hover,
.slideOuter .slide a#p4:hover {width:318px;}
```

6. To get the HTML code for the sliding menu, we go to `view source\view source`. This will prompt the following window:

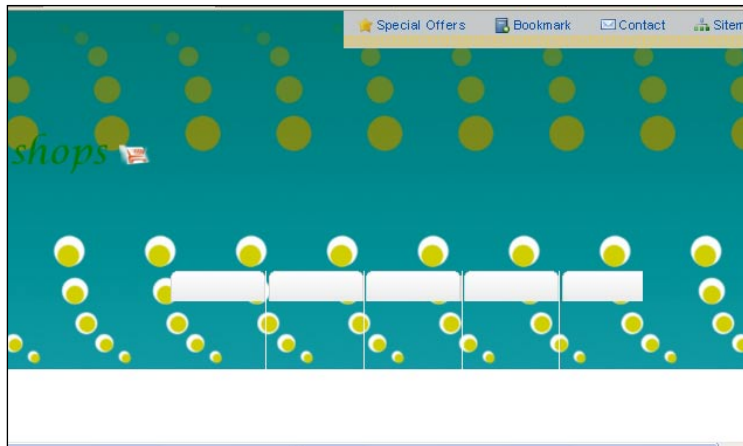


7. It will open up a new window containing the source code of the menu:

```
<div class="slideOuter">
 <ul class="slide current">

</div>
```

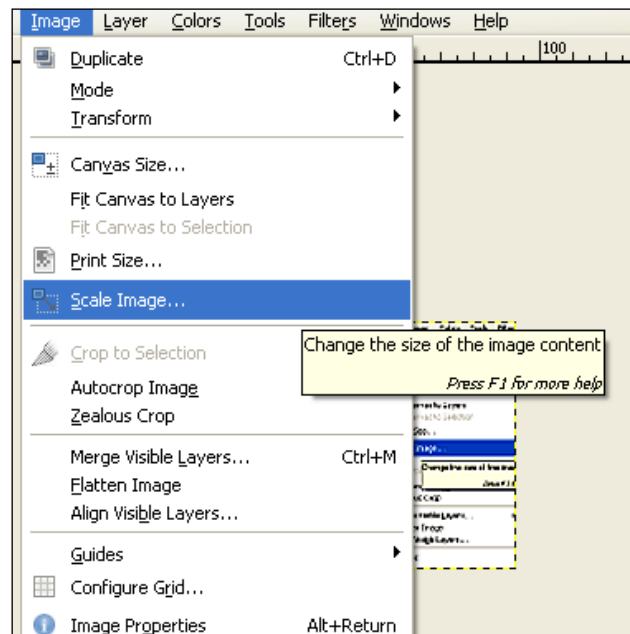
8. We will place this code in the `header.tpl` file under `catalog\view\theme\shop\template\common`. We are going to replace the existing code of `div4` class with the above code.
9. If we refresh our browser, we find the following state of our menu:



10. Now, we will create images for the menu. We will search the Internet for relevant images. First, let's check for the home tab. In this recipe, we will use the following image for the home tab:



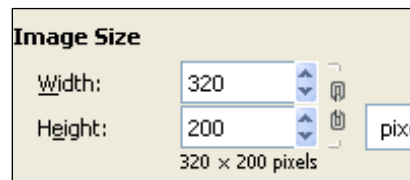
11. Open up the image with **GIMP**. We scale the image to **320x200**. You can scale to your desired size. Click on Image\Scale image. It will open the following window:



12. This will prompt a window where we can enter the image dimensions. First, we will unlink the link icon besides the width and height of the image;



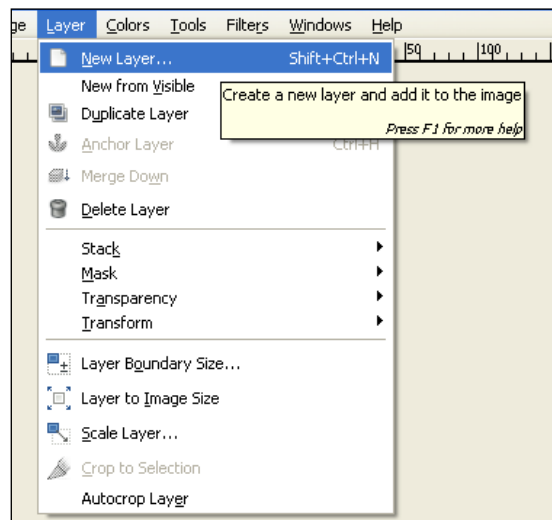
13. Now, we will enter the scaling height and width into the window:



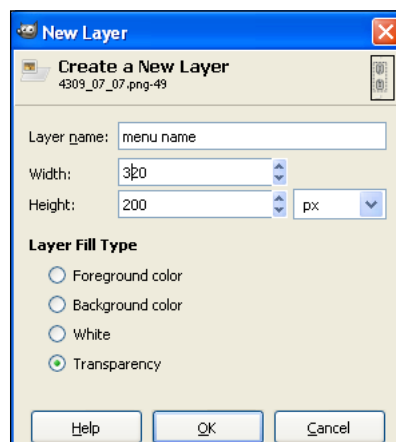
14. After clicking the **scale** button, our image will be converted to a new image. See the following image:



15. Now, we will create a layer on the image.



16. We will name it as **menu name** and set the height and width as **200px** and **320px**. And, the **Layer Fill Type** to **Transparent**.



17. We can view the layer window by selecting `windows\dockable dialogs\layers`. We will see our newly created layer there.
18. Now, choose the **rectangle select tool** from toolbox:



19. We select an area on the left-hand side of the image. See the following image. We select the rectangular box area of the image:



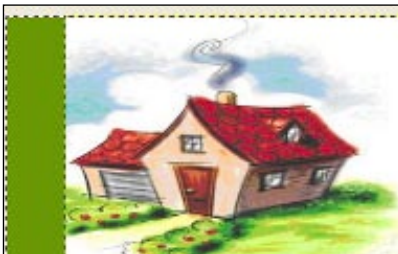
20. We choose `#5f8e03` from the color panel of the tool box.



21. Then, click the **Bucket Fill tool** from the tool box:



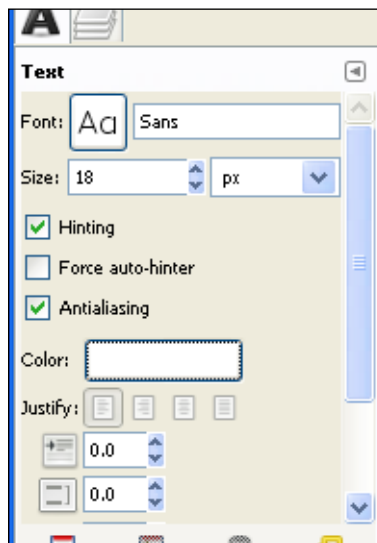
22. And, we apply the color on our selected area on the image:



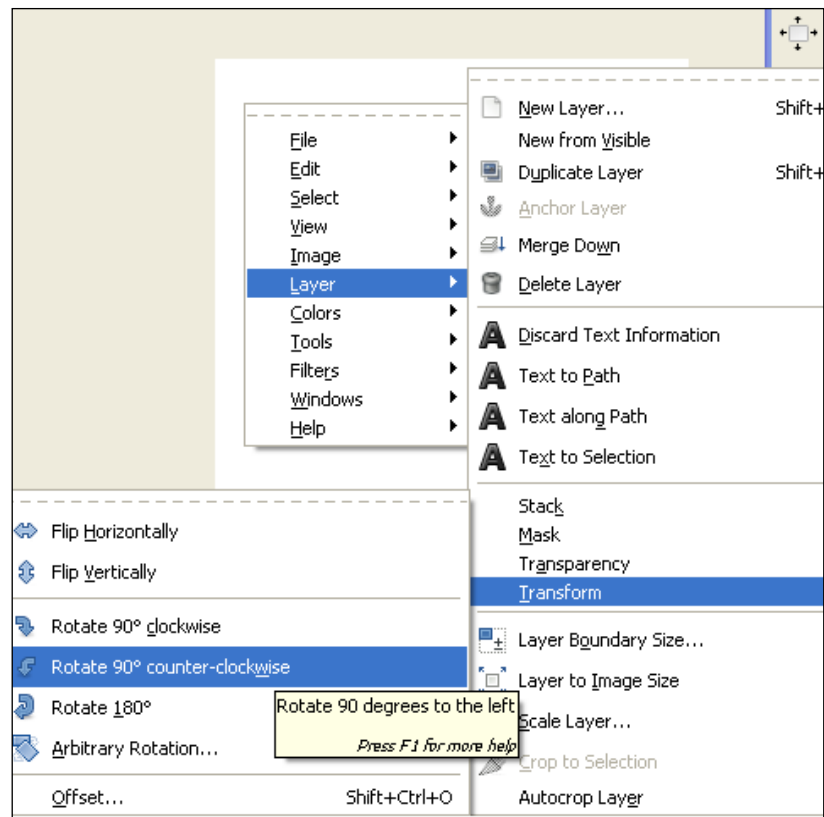
23. We will write some text on the select area of the layer. Choose the **Text Tool** from the tool box:



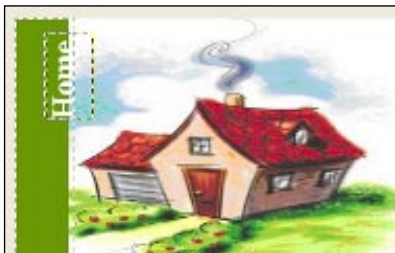
24. To open the option window for the **Text Tool**, go to windows\dockable dialogs\ fonts. Also, select windows\dockable dialogs\tool options. This shows the text option's window:



25. We select **Serif Bold** as the font, the font size as **25px**, and the color as **#ffffff**. We write **Home** as our text. Now, right click on the text layer and select **Layer | Transform | Rotate 900 counter clock-wise**.



26. This will change our image to this:



27. Then, we select the **move** tool:





28. We move the **text** layer to the following position:



29. We select the **smudge** tool from the tool box:



30. From the layer's option window, we will select the **menu name** layer. And then, we use the **smudge** tool on that layer. So, our image will become like this:



31. We will save it as `home-menu.png` under `catalog\view\theme\shop\image`. We need to create other tab images in a similar way.

For the **Login** tab, we created an image like this:



And the **Logout** tab gets the image as the following:



We created the following image for the **My Account** section:



We created the **Basket** tab as the following image:



The **Checkout** tab holds the following image:



32. So, we have our tab images. Now, we will place their path links in the **src** of the tab image. See the following code block and notice the value of the `src` attribute. Here, we place the path value of our images. Then, the horizontal slider menu will hold the following code block:

```
<div class="slideOuter">
 <ul class="slide current">

</div>
```

33. And if we refresh our browser, we will find our slider in the following state:



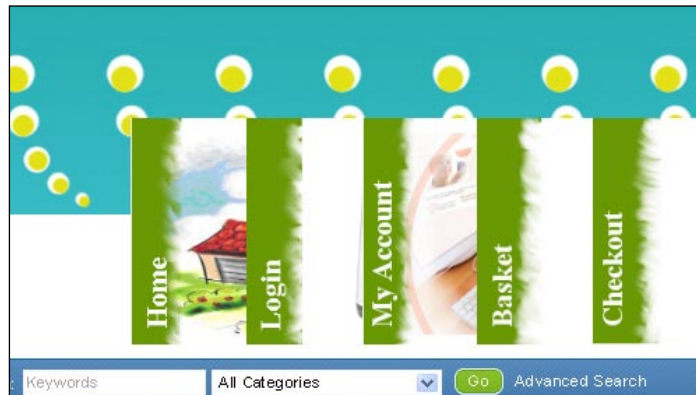
34. You will find that there is an extra area on the top of each of the tab images. We need to remove this. We will remove the `background` property for this style in `stylesheet.css` under `catalog\view\theme\shop\stylesheet`.

```
#header .div4 a {
//...
background: url('../image/tab_1.png') no-repeat;
//...
}
```

35. Also, we will remove `border` property from our `slideOuter` class style:

```
.slideOuter .slide li a {
//...
border-right:2px solid #fff;
//...
}
```

36. Then, our horizontal slider will become like this:



### How it works...

In the above recipe, we first get the style code, then we have created our images for the navigation and after that we place the path to our newly created images in our code block. We have explained each step and put relevant images there.

### There's more ...

You can also use the **slide tabs** menu. Go to the link: <http://www.cssplay.co.uk/menus/slide-tabs.html>.

## Adding a fisheye menu

In this recipe, we will add a popular fisheye menu into our store **shop**. We will create the images and change the codes.

### Getting started

We will add a menu like this link: <http://www.cssplay.co.uk/menus/fisheye8.html>

## How to do it

We will follow these steps:

1. Like the previous recipe, we will use a fisheye menu from the cssplay website. Go to this link: <http://www.cssplay.co.uk/menus/fisheye8.html>. We will use it in our shop.
2. We will use the Web developer tool to get the css style and related HTML code block for this fisheye menu.
3. We get the following css code style for the menu from the **View Source** option:

```
<style type="text/css">
/* =====
=
This copyright notice must be untouched at all times.
```

```
The original version of this stylesheet and the associated (x)html
is available at http://www.cssplay.co.uk/menu/fisheye8.html
Copyright (c) 2005-2009 Stu Nicholls. All rights reserved.
This stylesheet and the associated (x)html may be modified in any
way to fit your requirements.
```

```
=====
= */
#container {background:url(fisheye4/background.jpg); width:750px;
height:250px; padding-top:50px;}
#holder {width:700px; height:140px; margin:0 auto;}

ul.outer {padding:0; margin:0; list-style:none; width:140px;
height:140px; float:left;}
ul.outer ul {padding:0; margin:0; list-style:none;}
ul.outer a {text-decoration:none;}

ul.outer li {display:block; float:left; padding:3px; margin:0;}

ul.outer li.inner img.other {width:100%; display:block; border:0;
cursor:pointer;}
ul.outer li.inner img.ie6 {display:none;}

ul.outer li:hover {padding:1px;}

ul.outer li.inner:hover img {width:128px; height:128px;}
</style>
```

```

<!--[if lte IE 6]>
<style type="text/css">
ul.outer table {border-collapse:collapse; padding:0; margin:-1px;}

ul.outer li {padding:0;}
ul.outer li a {float:left; display:block; padding:3px;}
ul.outer li.inner img.other {display:none;}
ul.outer li.inner img.ie6 {width:100%; display:block; border:0;
cursor:pointer;}

ul.outer li.inner img#login {width:100%; background:none; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='fisheye4/
login.png', sizingMethod='scale');}
ul.outer li.inner img#camera {width:100%; background:none; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='fisheye4/
camera.png', sizingMethod='scale');}
ul.outer li.inner img#phone {width:100%; background:none; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='fisheye4/
phone.png', sizingMethod='scale');}
ul.outer li.inner img#find {width:100%; background:none; filter:
progid:DXImageTransform.Microsoft.AlphaImageLoader(src='fisheye4/
find.png', sizingMethod='scale');}
ul.outer li.inner img#printer {width:100%; background:none;
filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='fis
heye4/printer.png', sizingMethod='scale');}

ul.outer a:hover {padding:1px;}
</style>
<![endif]-->

```

4. We will put this code in either a new CSS file or we can put it in the header.tpl file under catalog\view\theme\shop\template\common. If we use a new CSS file, then we need to add the style file with the link attribute and also get the HTML code for it. We will place it under the div4 class of header ID of header.tpl file:

```

<div id="info">

<h2>Unusual - Fisheye image menu version 3 using just CSS</h2>
<h3>26th August 2009</h3>

<div id="container">
<div id="holder">
<ul class="outer"><!--[if lte IE 6]><table><tr>
<td><![endif]-->

```

```
<!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <li class="inner">

 <!-- [if lte IE 6]></td></tr></table><![endif]-->

 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->

<ul class="outer"><!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <!-- [if lte IE 6]><table><tr><td><![endif]-->
 <li class="inner">

 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
 <!-- [if lte IE 6]></td></tr></table><![endif]-->
->
```

```

 <!--[if lte IE 6]></td></tr></table><![endif]-->

<ul class="outer"><!--[if lte IE 6]><table><tr>
<td><![endif]-->
 <!--[if lte IE 6]><table><tr><td><![endi
f]-->

 <!--[if lte IE 6]><table><tr><td><![e
ndif]-->
 <!--[if lte IE 6]><table><tr><td><
![endif]-->
 <!--[if lte IE 6]><table><tr><t
d><![endif]-->
 <li class="inner">

 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table><![endif]-
->

 <!--[if lte IE 6]></td></tr></table><![endif]-->

<ul class="outer"><!--[if lte IE 6]><table><tr>
<td><![endif]-->
 <!--[if lte IE 6]><table><tr><td><![endi
f]-->
 <!--[if lte IE 6]><table><tr><td><![e
ndif]-->
 <!--[if lte IE 6]><table><tr><td><
![endif]-->
 <!--[if lte IE 6]><table><tr><t
d><![endif]-->
 <li class="inner">

 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table><![endif]-->
 <!--[if lte IE 6]></td></tr></table><![endif]-->


```



```
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table><![endif]-
->
 <!--[if lte IE 6]></td></tr></table><![endif]-->

<ul class="outer"><!--[if lte IE 6]><table><tr>
<td><![endif]-->
 <!--[if lte IE 6]><table><tr><td><![endi
f]-->
 <!--[if lte IE 6]><table><tr><td><![e
ndif]-->
 <!--[if lte IE 6]><table><tr><td><
![endif]-->
 <!--[if lte IE 6]><table><tr><t
d><![endif]-->
 <li class="inner">

 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table></
a><![endif]-->
 <!--[if lte IE 6]></td></tr></table><![endif]-
->
 <!--[if lte IE 6]></td></tr></table><![endif]-->

</div>
</div>

<p class="info">Copyright ©2009 stu nicholls - cssplay.
co.uk</p>

</div> <!-- end of info -->
```

5. We need to adjust some settings for this HTML code structure. We will add an `id` on the image tag for each of our menu items. So, the first ID will be `home` and the code will be as this:

```

```

6. In this way, we will add the other `ids` in the image tag. Now, we will place an image on the `src` attribute of the image. We will use the following image and scale height to **150px** and the width accordingly.



7. We create a `png` and `gif` image for each tab. So, our code becomes like this for the home ID of the image:

```

```

8. In a similar way, we will add other images to the remaining image tags and change the image `id` also.
9. We also need to add a condition to check whether a user is logged in or not. On the basis of that we will use login or logout. We can use the following code:

```
<?php if (!$logged) { ?>
 //...
 <?php } else { ?>
 //...
 <?php } ?>
```

10. Then, the slider will become like this:



11. Now, we need to adjust the HTML and CSS style of the menu. We remove the following code block:

```
<div id="info">
<h2>Unusual - Fisheye image menu version 3 using just CSS</h2>
<h3>26th August 2009</h3>
```

```


//...

<p class="info">Copyright ©2009 stu nicholls - cssplay.
co.uk</p>

</div> <!-- end of info -->
```

12. Also, we will remove the following styles from the css style:

```
#container {
background:url(fisheye4/background.jpg);
height:250px;
padding-top:50px;
}
```

13. When we refresh the browser, we will see the following changes:



14. Now, again, the following changes:

```
<div id="container">
//...
</div>
```

And change the style also:

```
#container {
width: 750px;
}
```

In the stylesheet.css file, we remove this:

```
#header .div4 {
width:510px;
}
```

15. So, we get the following image of our menu:



16. We also remove the background image for each div:

```
#header .div4 a {
 background:url("../image/tab_1.png") no-repeat scroll 0 0
 transparent;
}
```

17. The final status of the our menu becomes like the following:



### How it works ...

### There's more ...

You can also use the **html time flylist** menu. Go to the link: <http://www.cssplay.co.uk/menus/html-plus-time-flylist-menu.html>

### See also

- Displaying horizontal sliding menus

## Giving a background color to the menu

We have added horizontal slider and fisheye menus in the previous two recipes. In this recipe, we will add a total background menu to the store **shop**.

### Getting started

We will add a menu like the one shown in this link: [http://www.cssplay.co.uk/menus/total\\_background.html](http://www.cssplay.co.uk/menus/total_background.html)

### How to do it

We will follow these steps to integrate the total background menu into our site:

1. We will go to get the style code and HTML elements using **Web developer**. Click on the **View Source** button. We will bring the following style code for our **total background** menu:

```
/* =====
=
This copyright notice must be untouched at all times.

The original version of this stylesheet and the associated (x)html
is available at http://www.cssplay.co.uk/menus/total_background.html
Copyright (c) 2005-2009 Stu Nicholls. All rights reserved.
This stylesheet and the associated (x)html may be modified in any
way to fit your requirements.
=====
```

```
/*
#menu {width:750px; height:45px; margin:20px auto; background:
url(home.png) left top; position:relative;}
#menu ul {padding:0; margin:0; list-style:none; position:absolute;
left:0; top:0; width:750px; height:45px; z-index:10;}
#menu ul li {float:left; padding:0;}
#menu ul li#li1 {padding-left:50px;}
#menu li a {display:block; height:30px; float:left; color:#fff;
text-decoration:none; line-height:30px; padding:0 10px; font-
family:arial, sans-serif; font-size:14px; border-right:1px solid
#bbb; margin-top:5px;}
#menu li#li7 a {border:0;}

#menu li#li1 a b {background:url(home.png) no-repeat left 100px;}
#menu li#li2 a b {background:url(calculator.png) no-repeat left
100px;}
```

```
#menu li#li3 a b {background:url(calendar.png) no-repeat left
100px;}
#menu li#li4 a b {background:url(clipboard.png) no-repeat left
100px;}
#menu li#li5 a b {background:url(letters.png) no-repeat left
100px;}
#menu li#li6 a b {background:url(printer.png) no-repeat left
100px;}
#menu li#li7 a b {background:url(secure.png no-repeat) left
100px;}

#menu li a:hover {color:#000; white-space:nowrap;}
#menu li a.current {color:#ff0;}
#menu li a:hover.current {color:#ff0;}
#menu li a:hover b {display:block; width:750px; height:45px;
position:absolute; left:0; top:0; z-index:-1;}

#menu li#li1 a:hover b {background:url(home.png) left top;}
#menu li#li2 a:hover b {background:url(calculator.png) left top;}
#menu li#li3 a:hover b {background:url(calendar.png) left top;}
#menu li#li4 a:hover b {background:url(clipboard.png) left top;}
#menu li#li5 a:hover b {background:url(letters.png) left top;}
#menu li#li6 a:hover b {background:url(printer.png) left top;}
#menu li#li7 a:hover b {background:url(secure.png) left top;}
```

2. We will put this code in either a new css file or we can put it in the header.tpl file under catalog/view/theme/shop/template/common. If we use a new css file, then we need to add the style file with the link attribute and also get the HTML elements from there. We will place it under the div4 class of header ID of the header.tpl file:

```
<div id="info">

<h2>Unusual - Total background hover</h2>
<h3>20th April 2009</h3>

<div id="menu">

 <li id="li1">home</
a>
 <li id="li2">calculator
 <li id="li3">calendar
 <li id="li4">clipboard
```

```

 <li id="li5">correspondence

 <li id="li6">printer
 <li id="li7">secure area

</div>

<p class="info">Copyright ©2009 stu nicholls - cssplay.
co.uk</p>

</div> <!-- end of info -->

```

3. We remove the following code from the HTML structure:

```

<div id="info">

<h2>Unusual - Total background hover</h2>
<h3>20th April 2009</h3>

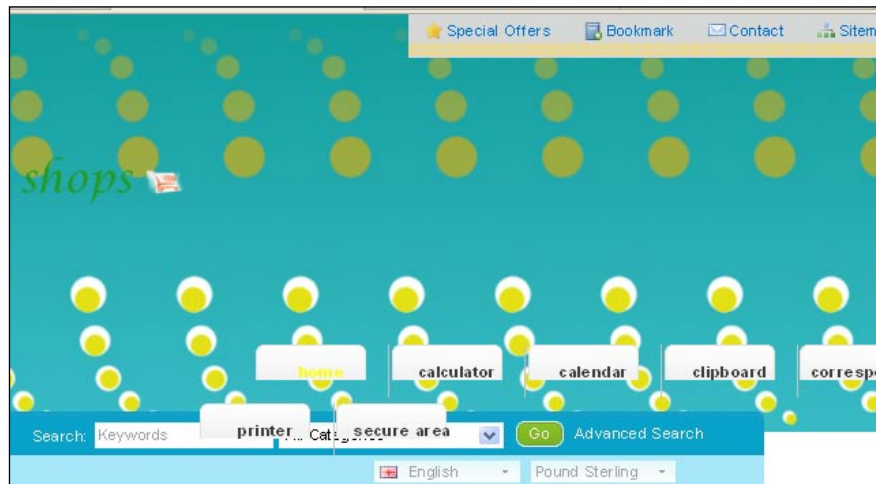
//...

<p class="info">Copyright ©2009 stu nicholls - cssplay.
co.uk</p>

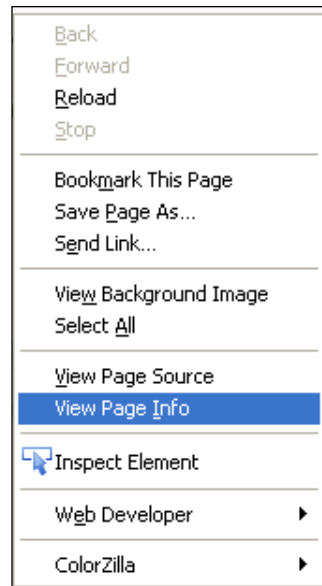
</div> <!-- end of info -->

```

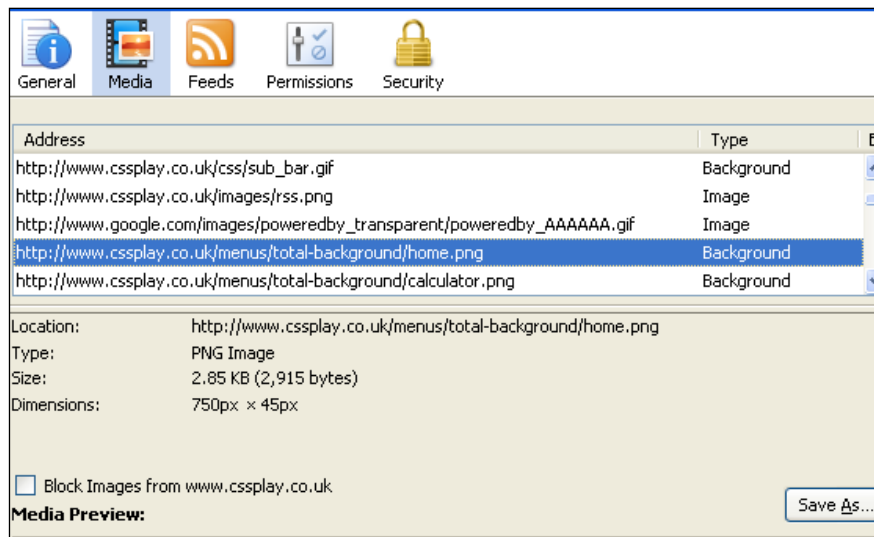
4. If we refresh our browser, we get the following status of our menu:



5. Now, we need to modify the images for our menu. We get the existing images from the link: [http://www.cssplay.co.uk/menus/total\\_background.html](http://www.cssplay.co.uk/menus/total_background.html).
6. Right click the mouse. Then, click **View page info**.



7. We go to the **Media** tab. It will display all the images on the page:





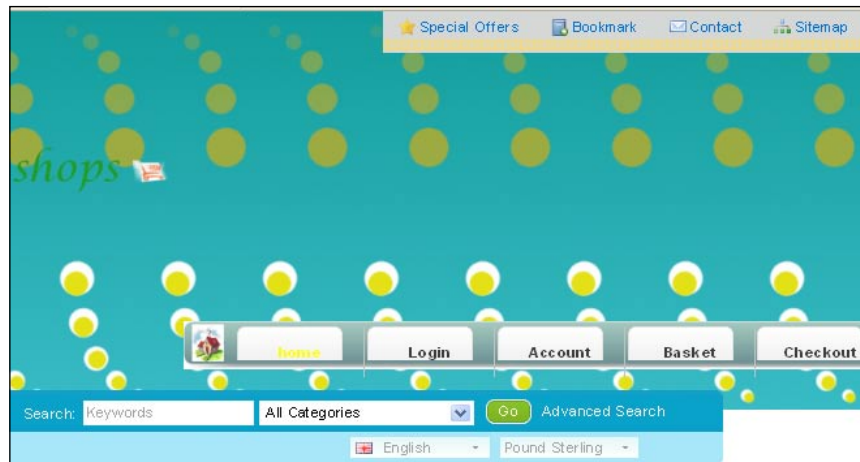
8. Click the **Save as** button. We get the other images in this way.
9. Now, we open up the image with GIMP. And also, open up our **login** menu item image in GIMP. We scale the image to **40x40** and then paste it into the **total background** image. We move the **login** icon on the left most side of the image. So, our image becomes like this:



10. We place the image at `catalog\view\theme\shop\image`. We need to change the location of the images in the style code for **total background** menu:

```
#menu {width:750px; height:45px; margin:20px auto; background:
url(catalog/view/theme/shop/image/home.png) left top; position:
relative;}
//...
#menu li#li1 a b {background:url(catalog/view/theme/shop/image/
home.png) no-repeat left 100px;}
//...
#menu li#li1 a: hover b {background:url(catalog/view/theme/shop/
image/home.png) left top;}
```

11. Our image becomes like the following:



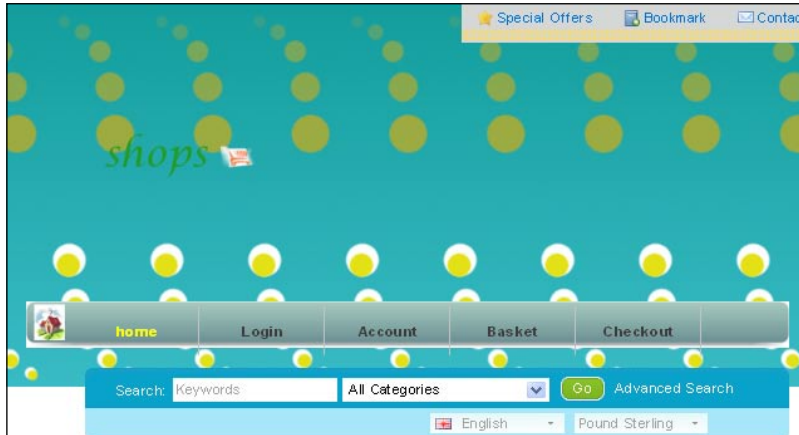
12. We remove the following style code from stylesheet under `catalog\view\theme\shop\stylesheet`:

```
#header .div4 a {
background:url("../image/tab_1.png") no-repeat scroll 0 0
transparent;
}
```

13. Also, remove the width attribute:

```
#header .div4 {
width:510px;
}
```

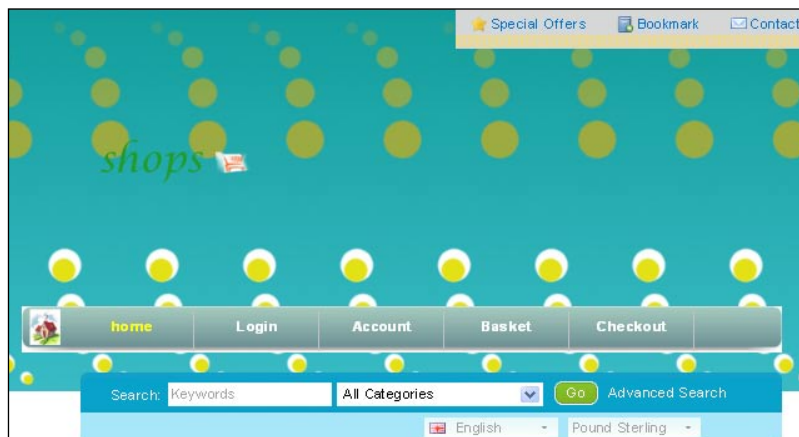
14. Now, we will go to the browser to view the changes we made:



15. Then, we adjust the position of the menu label. For this, we remove the following code block from the stylesheet:

```
#header .div4 a {
color:#333333;
padding-top:9px;
}
```

16. So, our menu becomes:



## There's more ...

You can also have the **dropdown** menu. Go to the link: <http://www.cssplay.co.uk/menus/css3-dropdown-all.html>.

## Integrating the social networking menu

We will add a social networking style menu in this recipe. This will be a circular menu. We will create our menu images and also adjust some styles for them.

## Getting started

Here, we also use a menu from [cssplay.co.uk](http://www.cssplay.co.uk/menus/css3-social-network.html). We will add a menu like this link: <http://www.cssplay.co.uk/menus/css3-social-network.html>.

## How to do it

We have to go through the following steps to have the social networking menu in our store **shop**:

1. We will have the style code and HTML element using Web developer's **view source**. Following is the style code:

```
ul.topul, ul.topul ul {padding:0; margin:0; list-style:none;
position:relative;}
ul.topul ul {position:absolute; left:-9999px; background:
url(trans.gif); opacity:0;
-webkit-transition: opacity 0.5s ease-in-out;
-moz-transition: opacity 0.5s ease-in-out;
-o-transition: opacity 0.5s ease-in-out;
transition: opacity 0.5s ease-in-out;
}
ul.topul {margin:50px auto; width:80px; height:80px;}
ul.topul ul li {width:48px; height:48px; float:left; display:
inline;}
ul.topul li.topli {width:80px; height:80px; float:left;}
ul.topul :hover ul {left:-48px; top:-48px; width:176px;
height:176px; opacity:1;}
ul.topul ul li img {width:48px; height:48px; display:block;
border:0; position:relative; left:0; top:0;
-webkit-transition: 0.2s ease-in-out;
-moz-transition: 0.2s ease-in-out;
-o-transition: 0.2s ease-in-out;
```

```

transition: 0.2s ease-in-out;
}
ul.topul li.p1 {margin-left:16px; margin-top:16px;}
ul.topul li.p3 {margin-top:16px;}
ul.topul li.p5 {margin-left:80px;}
ul.topul li.p6 {margin-left:16px;}
ul.topul li.p7 {margin-top:16px;}
ul.topul :hover ul li a:hover {position:relative; z-index:100;}
ul.topul li:hover ul li a:hover img {width:64px; height:64px;
left:-8px; top:-8px; z-index:100;}

/* for IE6 */
ul.topul table {border-collapse:collapse; position:absolute;
left:0; top:0; margin:-4px;}
ul.topul li a:hover {direction:ltr;}

```

2. `ul.topul li a:hover ul li a:hover img` {position:absolute; width:64px; height:64px; left:-8px; top:-8px; z-index:100;}
3. We will put this code in either a new css file or we can put it in `header.tpl` file under `catalog\view\theme\shop\template\common`. If we use a new css file, then we need to add the style file with the link attribute. And the HTML code is the following. We will place it under the `div4` class of header ID of the `header.tpl` file:

```

<div id="info">

<h2>CSS3 - The future now - 'Social Networking Menu' </h2>
<h3>28th September 2010</h3>

<ul class="topul">
 <li class="topli" id="s1"><!--[if lte IE 6]><table><tr><td><![endif]-->

 <li class="p1">
 <li class="p2">
 <li class="p3">
 <li class="p4">
 <li class="p5">
 <li class="p6">


```

```

 <li class="p7">
 <li class="p8">

 <!-- [if lte IE 6]></td></tr></table><![endif]-->

<a href="http://twitter.com/home?status=Currently testing a new
gallery from @stucssplay here http://www.cssplay.co.uk/menus/css3-
social-network.html" rel="nofollow" class="tweet">
<p class="info">Copyright ©2010 stu nicholls - CSS play</p>

</div> <!-- end of info -->

```

4. We resize our menu images to 50x50. The home tab image becomes like this:



5. We place the image at catalog\view\theme\shop\image. Now, we remove some redundant HTML elements from the code. We remove the following:

```

<div id="info">
<h2>CSS3 - The future now - 'Social Networking Menu' </h2>
<h3>28th September 2010</h3>
//...

<a href="http://twitter.com/home?status=Currently testing a new
gallery from @stucssplay here http://www.cssplay.co.uk/menus/css3-
social-network.html" rel="nofollow" class="tweet">
<p class="info">Copyright ©2010 stu nicholls - CSS play</p>

</div> <!-- end of info -->

```

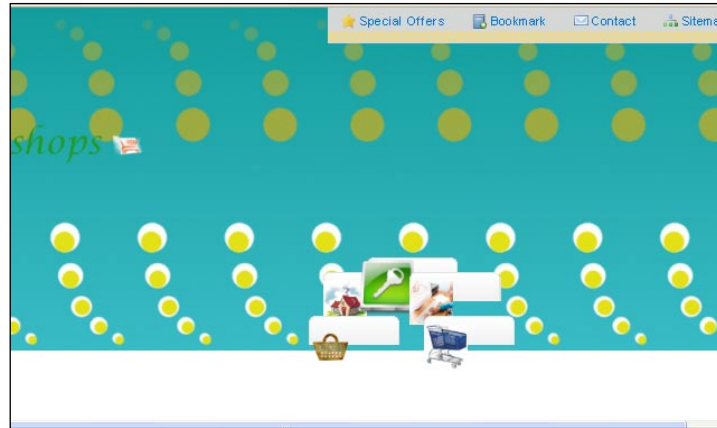
6. You need to change the image path in the menu HTML element:

```

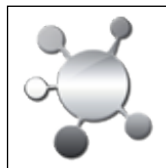
<li class="p1">

```

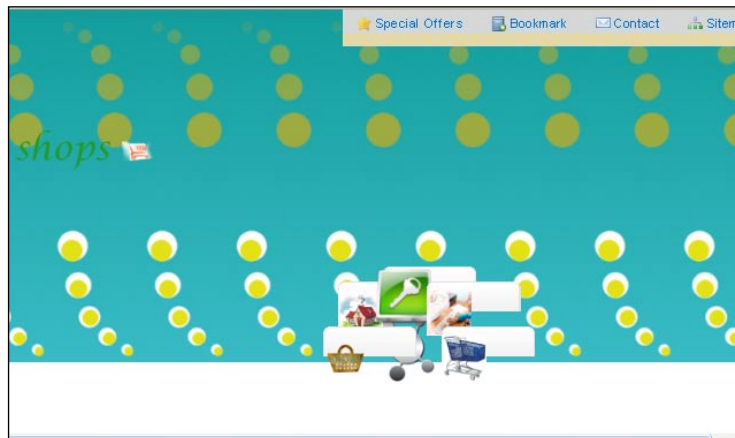
7. Then, it becomes as the following:



8. We will download the `social.png` image using **View page info** | **Media**.



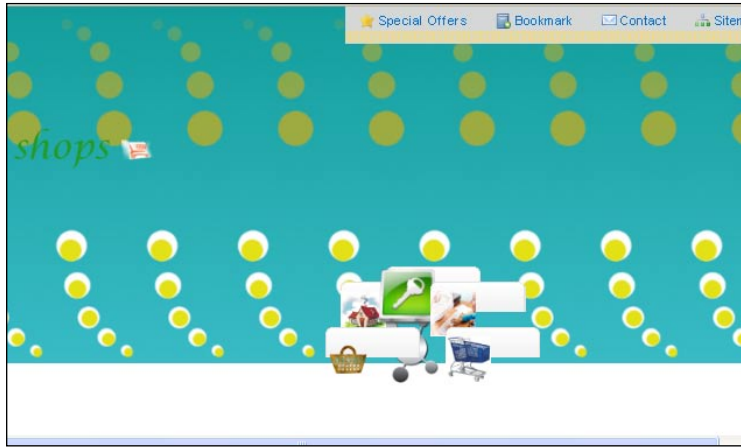
9. So now, we will refresh the browser:



10. We remove some style attributes from stylesheet:

```
#header .div4 a {
background:url("../image/tab_1.png") no-repeat scroll 0 0
transparent;
}
```

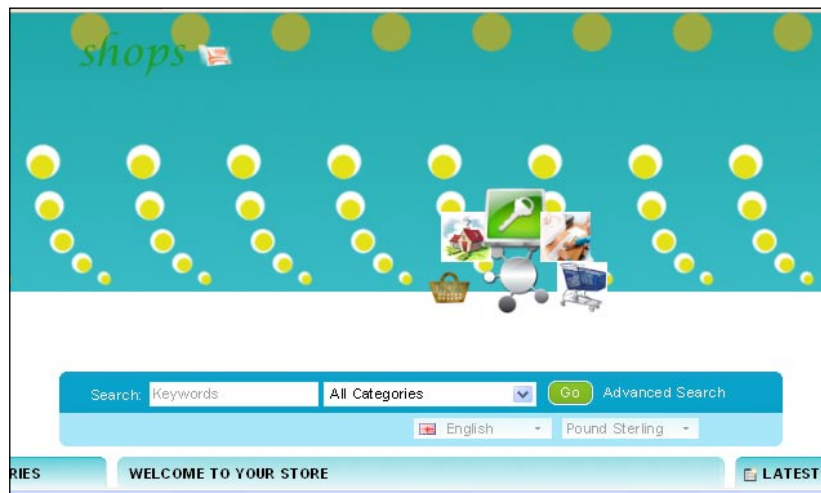
11. When we go the browser, then our menu becomes like this:



12. We have to change some style code to adjust the placement of our social networking menu. We remove the following style code:

```
#header .div4 a {
padding-top:100px;
}
```

13. Then, our menu will be like the following:



## There's more...

You can also integrate the **rising tab** menu. Go to the link: <http://www.cssplay.co.uk/menus/css3-rising-tabs.html>.

# 8

## Footer Layout

In this chapter, we will cover:

- ▶ Creating wide area footers
- ▶ Implementing a footer with a gradient image
- ▶ Addition of the three column footer

### Introduction

This chapter will show us how to create attractive footers. We will see a step-by-step development of our footers. Here, we will discuss every step with the proper code and relevant images. So, it will be an easy ride in the footer creation world.

This chapter will guide us in developing different attractive modern footers.

### Creating a wide footer for our store

First, we will create a wide footer for our store **shop**.

#### Getting started

Open up the `footer.tpl` file under `catalog\view\theme\shop\template\common` with our favourite editor.



## How to do it

We will follow these steps:

1. We will open the default `footer.tpl` file under `catalog\view\theme\shop\template\common`. We will replace the default footer ID with the new footer content. There are two div tags: `div1`, `div2`. Following is the default footer content:

```
<div class="div1"></div>
<!--
OpenCart is open source software and you are free to remove the
Powered By OpenCart if you want, but it's generally accepted
practise to make a small donation.
Please donate via PayPal to donate@opencart.com
//-->
<div class="div2"><?php echo $text_powered_by; ?></div>
<!--
OpenCart is open source software and you are free to remove the
Powered By OpenCart if you want, but its generally accepted
practise to make a small donatation.
Please donate via PayPal to donate@opencart.com
//-->
</div>
```

We will apply a background style in the footer ID. But first, we will create a background image for our footer.

2. Open GIMP and create a 60x60 image. We will fill the image with the color `#0d9492`. Then, we select the bucket fill tool from the toolbox:

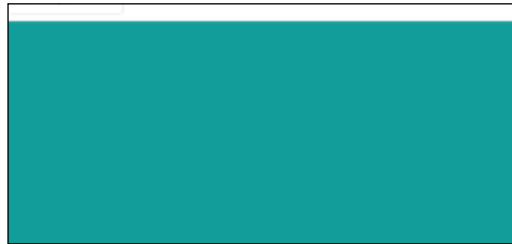


3. So, our background image becomes like this:



4. We place the following style code for the footer id in `stylesheet.css` under `catalog\view\theme\shop\stylesheet`:
 

```
#footer {
height: 500px;
background:transparent url(catalog/view/theme/shop/image/footer-
back.png) repeat top center;
margin-top:40px;
}
```
5. When we go to the browser, the footer will appear as the following:



6. We will have an inner div class in our opened `footer.tpl`:
 

```
<div class="inner">
//...
</div>
```
7. And apply the following style to this class in `stylesheet.css`:
 

```
#footer .inner{
padding:90px 20px 0px;
}
```
8. We add an inner block for site's information in the footer within the footer ID:
 

```
<div class="inner-block">
 <h3 class="widgettitle">Information</h3>

 Text one
 Text second</
a>
 Text third</
li>
 Text fourth</
a>
 Text fifth</
li>

</div>
```

```
<div class="inner-block">
 <h3 class="widgettitle">Information</h3>

 Text one
 Text second</
a>
 Text third</
li>
 Text fourth</
a>
 Text fifth</
li>

</div>
```

9. Now, we need to do some styling with it. We use the following style in our `stylesheet.css`:

```
.inner-block {
float: left;
width: 300px;
}
.inner-block ul {
list-style-type: none;
padding: 5px;
}
```

10. So, our footer becomes like this:



11. And we make a style for the **li** tags also in our stylesheet .css:

```
.inner-block ul li {
padding: 10px 0px;
border-top: 1px solid #7cbfbe;
}
```

12. This makes our footer the following:

Information	Information
Text one	Text one
Text second	Text second
Text third	Text third
Text fourth	Text fourth
Text fifth	Text fifth

13. We then apply a hover style for the li tag in our stylesheet .css:

```
.inner-block ul li:hover {
background-color: #0db492;
}
```

14. By hovering the li tag, we find the following changes:

Information	Information
Text one	Text one
Text second	Text second
Text third	Text third
Text fourth	Text fourth
Text fifth	Text fifth

15. Now, we will change the font colour of the **li** tag element in our `stylesheet.css`:

```
.inner-block ul li a {
padding: 0px 10px;
text-decoration: none;
color: #fff;
}
```

16. So, the footer style becomes like this:

Information	Information
Text one	Text one
Text second	Text second
Text third	Text third
Text fourth	Text fourth
Text fifth	Text fifth

17. We now set a hover effect for our **li** tag element:

```
.inner-block ul li a:hover {
color: #FFC62F;
}
```

18. When we hover the texts of **li** element, then the style appears as this:

Information	Information
Text one	Text one
Text second	Text second
Text third	Text third
Text fourth	Text fourth
Text fifth	Text fifth

19. Now, we add another block to the footer. We will use a contact us form here. We will add the following code block beside the information block:

```
<div class="widget webtreats_contact_form_widget" id="contact-
form-widget">
 <h3 class="widgettitle">Email Us</h3>

 <form method="post" id="contactFormWidget" action="">
 <p><input type="text" tabindex="5" size="22"
class="requiredField textfield" value="" id="contactNameWidget"
name="contactNameWidget"><label for="contactNameWidget"
class="textfield_label">Name *</label></p>

 <p><input type="text" tabindex="6" size="22"
class="requiredField email textfield" value="" id="emailWidget"
name="emailWidget"><label for="emailWidget" class="textfield_
label">Email *</label></p>

 <p><textarea class="requiredField textarea"
tabindex="7" cols="19" rows="20" id="commentsTextWidget"
name="commentsWidget"></textarea></p>
 <p><input type="submit" value="Submit"
tabindex="8" class="button in_footer" id="submittedWidget"
name="submittedWidget"></p>

 </form>
</div>
```

Information	Information	Email Us
Text one	Text one	<input type="text"/>
Text second	Text second	<input type="text"/>
Text third	Text third	<input type="text"/>
Text fourth	Text fourth	
Text fifth	Text fifth	

20. We need to add some styling to our contact us form. We add the following style elements here:

```
.webtreats_contact_form_widget h3{
margin-bottom:8px;
}
```

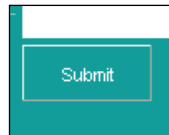
```
textarea#commentsTextWidget {
padding-left:8px;
height:107px;
overflow:hidden;
}
```

21. After applying this style, our footer becomes like this:

22. We also use a background style for buttons:

```
#footer .button{
background:transparent none;
height:42px;
width:97px;
}
```

23. So, our **Submit** button of the contact us form becomes like this:



24. We apply some padding and margin to the input fields:

```
#footer input#emailWidget, #footer input#contactNameWidget{
height:24px;
padding:5px 8px;
width:178px;
}
```

25. When we go to the browser, it brings the following changes:

26. Again, we make a background color change of the input fields. We add this css style to our code:

```
.webtreats_contact_form_widget label{
vertical-align:top;
}

.inputError ,input[type="text"].inputError {
border:1px solid #888;
}

#footer input[type=text], #footer textarea,#footer table, #footer
td, #footer th, #footer tr:hover{
background:#3ab3b2;
}
```

27. This makes our footer look like the following:



## Footer with a gradient image

We have seen a wide footer in our previous recipe. In this recipe, we will create another footer with a gradient image as the footer background. We will add the contact information, telephone numbers, and some useful links in the footer.

### Getting started

We open up the `footer.tpl` file under `catalog\view\theme\shop\template\common` with our favourite editor.

### How to do it

We will follow these steps:

1. We first create a footer container in the footer ID of our HTML block. The containing code is placed below:

```
<div id="footer">
 <div class="footer-wrap">
 <div class="footer-content">
 //...
 </div>
 </div>
</div>
```

2. For the footer ID, we will create a background image with the dimension **2x247**. We choose **#c6a50c** as the foreground colour and **#987e04** as the background color:



3. Now, we will choose the **gradient** tool from the tool box of GIMP:



4. We apply the selected colours with the **gradient** tool from top to bottom. We save the image as `footer.png`. We use some styling for our footer block in our `stylesheet.css`:

```
/* Footer */
#footer {
```

```

border-top: 1px solid #d0d3d8;
padding: 0.5em 0;
clear: both;
background: transparent url('catalog/view/theme/shop/image/
footer.png') repeat-x top left;

Min-height:246px;
width:100%;
margin:auto;
color: #fc0;
line-height:20px;
}

```

5. If we refresh our browser, we get our footer as this:



6. Our first container block will be the email block. We add the following code block there:

```

<div class="email">

 <p>admin@fishshop.com</p>
 <p>info@ fishshop.com</p>

</div>

```

7. So, the footer appears as follows:



8. We will add an image icon for the email block. We searched the Internet and chose the following image. You can find your favorite one.



9. And apply the following code in our footer.tpl:

```



```

10. Then, the footer becomes like this:



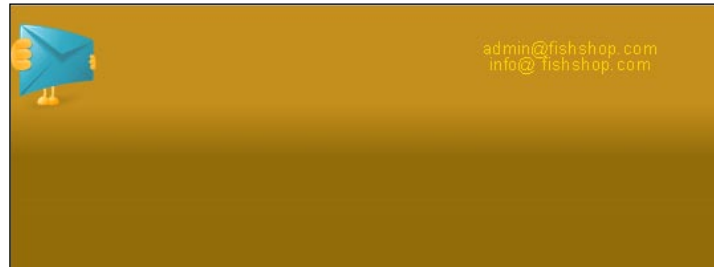
11. Our email block is at the top left corner, we will do some styling with it. See the following styles:

```
.footer-wrap{
 width:960px;
 margin:auto;
}

.footer-content{
 width:960px;
 margin:auto;
 overflow:hidden;
 padding-top:10px;
}

.footer-content p{
 font-size:16px;
 text-align: center;
}
```

12. This makes our email block more stylish. See the following image:



13. We add a **width** property for the email block. Here is the width code:

```
.email{
 width:330px;
 float:left;
}
```

14. Now, the email block appears like this:



15. We now add the phone block to the footer:

```
<div class="contact">

 <p>1.646.393.7800</p>
 <p>880197<span class="phone-word1" style="display:
inline;">fishshop.ltd</p>

</div>
```

16. In the browser, the footer becomes:



17. We also add another image icon for the phone block. We get the following image icon:



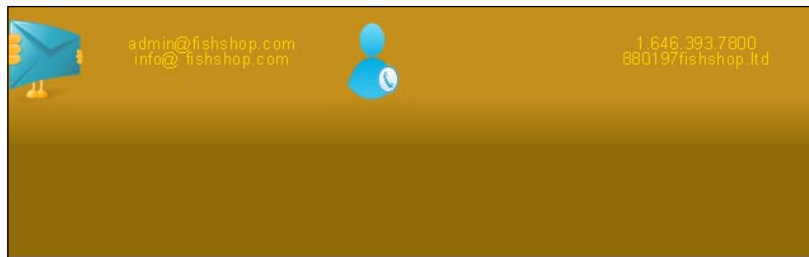
18. And add the following code, we can also use the css background property to use icons:

```



```

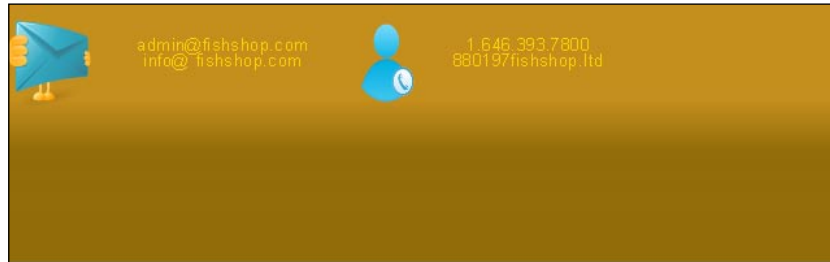
19. When we refresh our browser, it looks like this:



20. Now, we add the following style code block for the contact section:

```
.contact{
 width:270px;
 float:left;
}
```

21. This makes our footer look like the following:

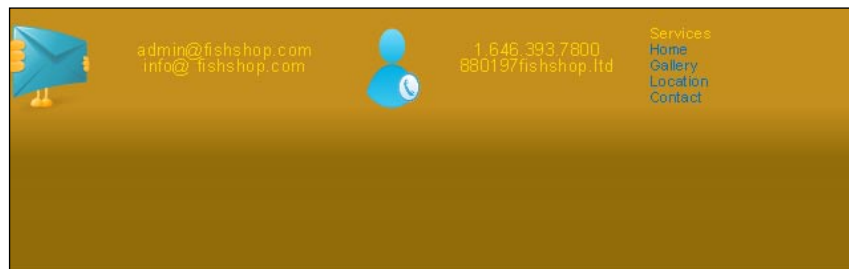


22. Now, add the **Services** block into our footer section. Here is the HTML code block for that:

```
<div class="footer-info">
 Services
 Home
 Gallery
 Location
 Contact

</div>
```

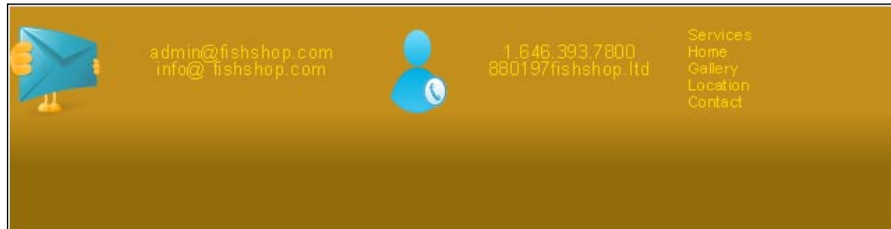
23. In the browser, we found it as this:



24. We need to do some styling to our list items for the services block:

```
#footer li a{
 color: #fc0;
 text-decoration:none;
}
```

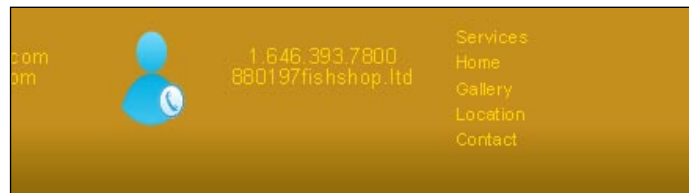
25. This brings the following changes to our site footer:



26. There needs to be some spacing between the footer links. So, we apply some padding there too:

```
.footer-info ul li{
 padding: 5px 5px 0px;
}
```

27. When we go to the browser, it looks like this:



28. We apply a different colour for the services block header.

```
.footer-info span{
 padding-bottom:2px;
 color:#fff;
}
```

29. It makes the services header look like this:



30. You can apply a bottom border to the service block header. For this, we will apply the following style:

```
.footer-info span{
 border-bottom:1px solid #ffffff;
}
```

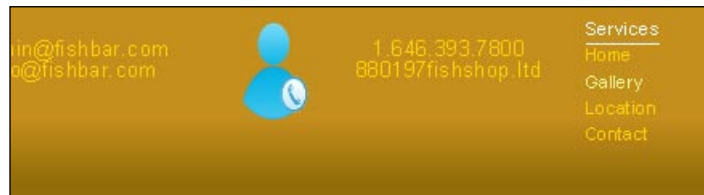
31. It brings a bottom border to the service block header.



32. We use a hover colour for the service block texts. We add the following style code for this purpose:

```
#footer li a:hover{
 color: #f7f888;
 text-decoration:none;
}
```

This brings about the following changes to the footer:



33. This completes the service block styling. We now add another block site-info to the footer:

```
<div class="footer-info">
 Site Info
 Privacy
 Terms & condition
 FAQ

</div>
```



34. Let's refresh our browser, we will find our footer as follows:



35. There will be a float property for the **site-info** block:

```
.footer-info{
 float: left;
}
```

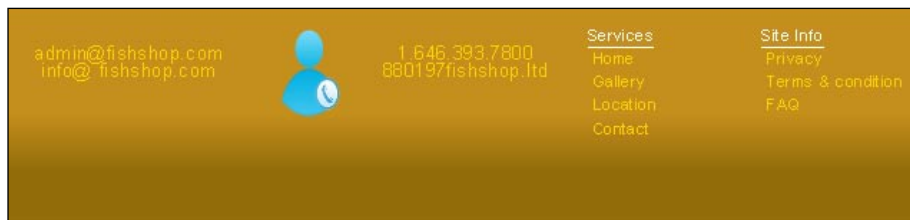
36. The blocks now become like the following:



37. Still the blocks are too close to each other. So, we apply a width for the information blocks:

```
.footer-info{
 width: 150px;
 float: left;
}
```

38. So, our footer now displays the following contents:



39. Now, place a divider between each section of the footer. We can also use the border colour to create a divider effect:

```
<div class="foot-divider"> </div>
```

We add the following styling for this:

```
.foot-divider{
height:130px;
width:1px;
margin:0px 20px;
float:left;
background:#c1c0b9;
}
```

40. In the browser, our footer now is like the following image:



41. At the bottom, we add the copyright section at the footer. Following is the HTML code block:

```
<div class="footer-bottom">
 <p class="copyright">Copyright © 2010 Fishshop.com.
All rights reserved.
 </p>
 <p class="copyright"> developed by: ABCD</p>
</div>
```

42. The footer now appears with the copyright section:



43. For the bottom class, we apply the following style code:

```
.footer-bottom{
 text-align:center;
 margin:auto;
 padding-bottom:40px;
 width:620px;
}
```

44. This makes our footer bottom like this:



45. We will add a site logo on the footer page. For this, we need to have the following `img` tag:

```

```

46. We can view the footer with our site logo now:



## Creating a three column footer

In this recipe, we will learn how to create a three column footer. We are going to add each column step by step.

### Getting started

We bring the footer.tpl file under Catalog\view\theme\shop\template\common with our favourite editor.

### How to do it

We will follow these steps:

1. Under the footer ID, we create our first column. We apply the following code block for this column:

```
<div class="column column1">
 <div id="community_news-2">
 <h3 class="widgettitle">Site Features</h3>
 <div class="entry box_entry footer-box">
 <h4><a href="#" rel="bookmark"
title="Permanent Link: Apple rebrands to Banana!">Enjoy Ease and
Convenience</h4>
 <a class="alignleft preloading_
background" href="#" title="Apple rebrands to Banana!">
 <p>Manage your account online anytime,
from anywhere. View and print your bills in seconds. Relax with
email alerts and payments information.</p>
 </div>

 <div class="entry box_entry footer-box">
 <h4><a href="#" rel="bookmark"
title="Permanent Link: Hello world!">Save Time and Money</h4>

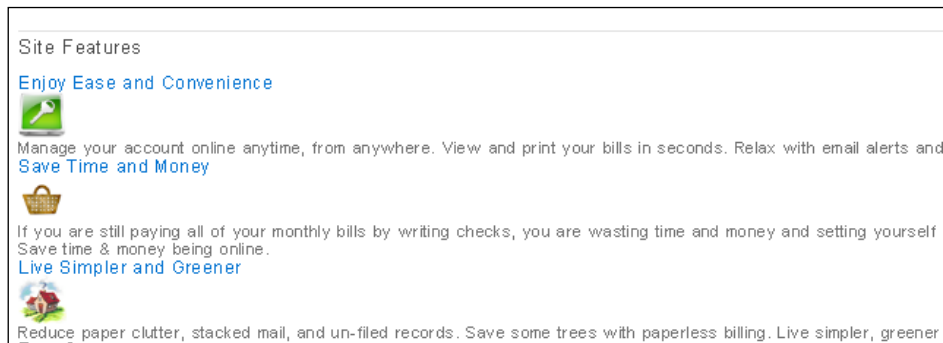
 <a class="alignleft preloading_
background" href="#" title="Hello world!">
 <p>If you are still paying all of your
monthly bills by writing checks, you are wasting time and money
and setting yourself up for potential late fees.
 Save time & money being online.
```

```

 </p>
 </div>
 <div class="entry box_entry footer-box">
 <h4><a href="#" rel="bookmark"
title="Permanent Link: Adobe Wallpaper contest">Live Simpler and
Greener</h4>
 <a class="alignleft preloading_
background" href="#" title="Adobe Wallpaper contest">
 <p>Reduce paper clutter, stacked mail,
and un-filed records. Save some trees with paperless billing.
 Live simpler, greener and clean.
 </p>
 </div>
</div>
</div>
</div>

```

- Without footer style, our footer now appears as this:



- Now, we apply a style code for our footer in our `stylesheet.css`. We will make our footer a little wider. For IE 6 fixes, we can use the styles for those fixes in a different style file:

```

#footer{
color: #333;
background: #DFA200;
position:relative;
width:1000px;
margin:0 auto;
padding-left:0px !important; /*ie6fix*/
padding-left:20px;
z-index:0;

```

```
padding-top:40px;
clear: both;
}
```

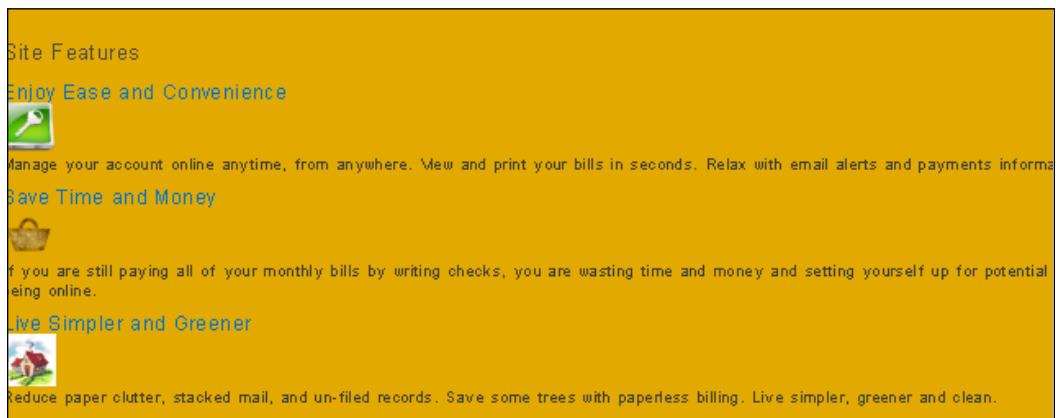
4. After applying the style, our code becomes as this:



5. For the `box_entry` class, we apply style for font size, line-height, etc:

```
.box_entry{
font-size: 11.5p;
line-height: 1.6em;
padding:0;
margin: 0 0 10px 0;
}
```

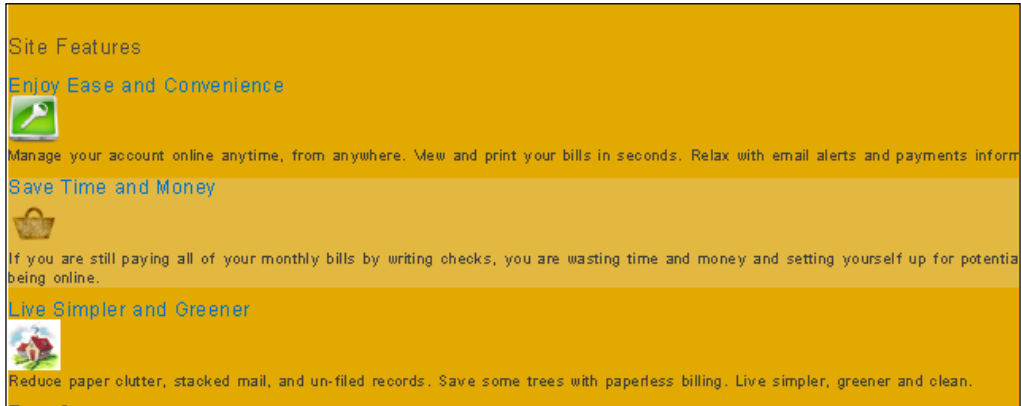
6. This brings the following changes to our footer:



7. We apply hover effect for this column with the following code:

```
#footer .box_entry:hover {
background-color: #DFB137;
}
```

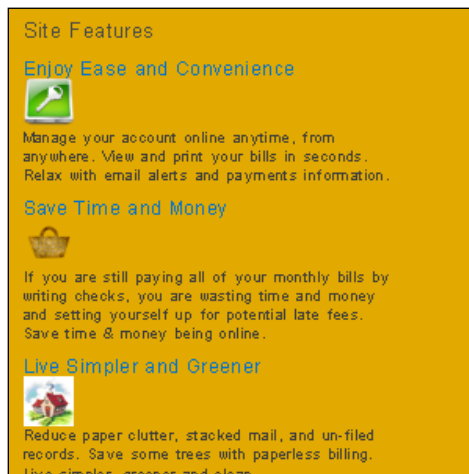
8. This brings about the following changes:



9. Now, we resize the first column. We apply a width to it:

```
.column {
width:293px;
padding-left:20px;
margin-right:20px;
}
```

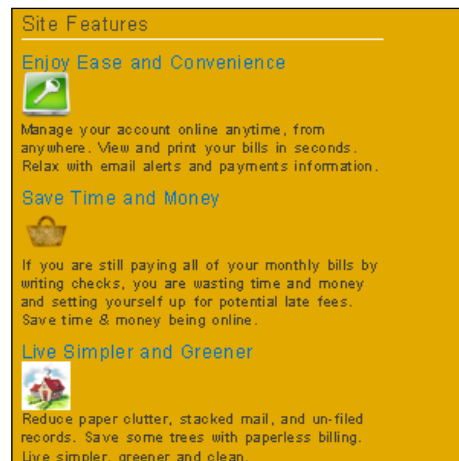
10. When we go to the browser, we find our footer as this:



11. We have a bottom border at the footer header. So, we add this code for the header title:

```
#footer h3, #top #footer .box_entry{
 border-bottom:1px solid #fff;
 padding-bottom: 5px;
}
```

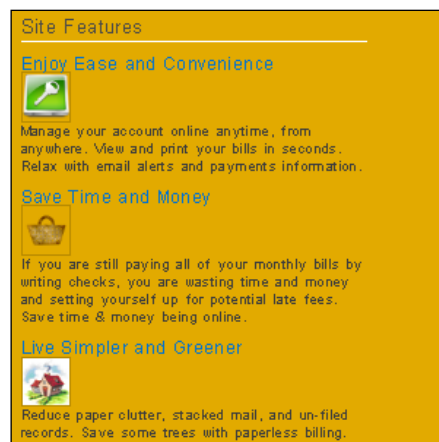
12. Refresh the browser and bring the changes:



13. And apply a border to the footer images:

```
#footer .footer-box img{
 border:1px solid #AF7F00;
}
```

14. This brings the following changes:

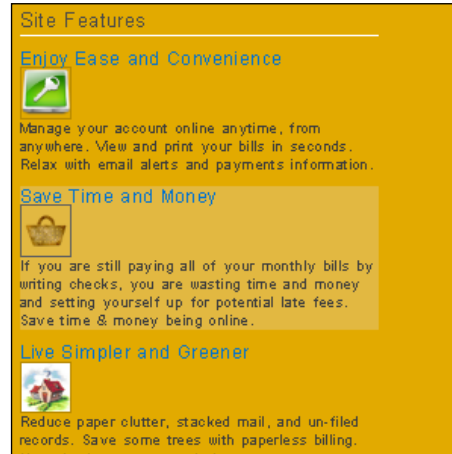




15. We apply a hover effect to our image border style:

```
#footer .footer-box img:hover{
 border:1px solid #555;
}
```

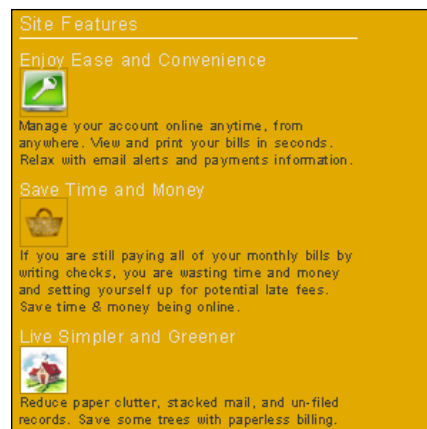
16. When we hover the images, the style changes like:



17. For the titles and section headers, we change the font colour to this:

```
#footer h1, #footer h2, #footer h3, #footer h4, #footer h5,
#footer h6, #footer legend, #footer h1 a, #footer h2 a, #footer h3
a, #footer h4 a, #footer h5 a, #footer h6 a{
 color: #ddd;
}
```

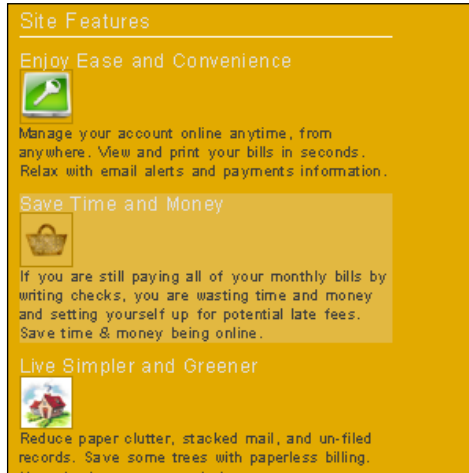
18. This brings the following change:



19. For hover effect with our section title, we use this code block:

```
#footer h4 a:hover{
 color: #999;
}
```

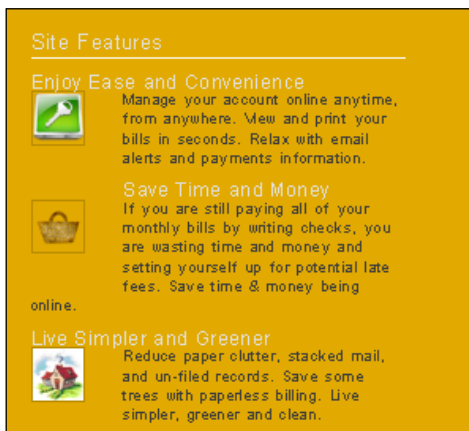
20. When we hover the title, it brings the changes:



21. Now, we add this code style:

```
.alignleft{
 float:left;
 margin:0px 15px 15px 0px;
 display: block;
}
```

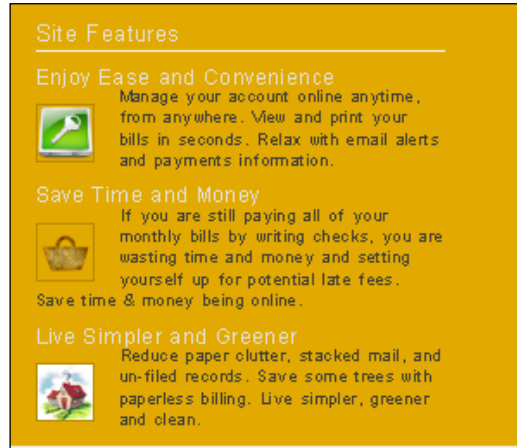
22. This brings the first column as the following:



23. We use a margin property to the first column:

```
.box_entry .alignleft{
margin:6px 9px 0px 0;
}
```

24. And our column appears as the following:



25. Now, we add column two to our footer. First, we add the HTML code style:

```
<div class="column column2">
 <div class="footer-box">
 <h3>Benefits</h3>
 <p>Grameen billing system is the unique
way to keep you informed about your bills. Using online billing
system you have several benefits.</p>
 <p class="small_block">You don't need to
keep paper copy of your bills.</p>
 <p class="small_block">Pay bills from
anywhere.</p>
 <p class="small_block">Track your expense
that you have spent.</p>
 <p class="small_block">Access your account
from being online.</p>
 </div>
</div>
```

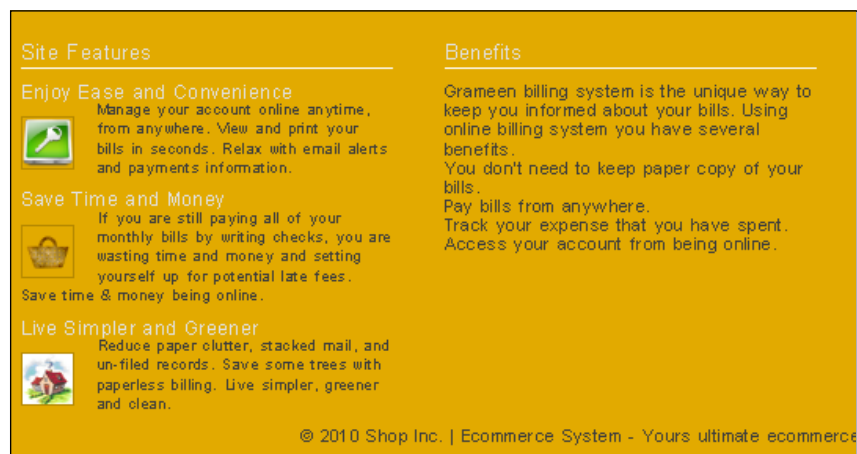
26. With two columns, the footer appears as this:



27. We apply a float attribute to the column class:

```
.column {
float:left;
}
```

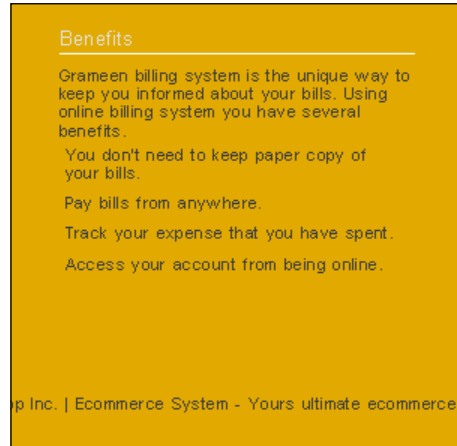
28. This brings the two columns side by side:



29. You need to add padding to the `small_block` class:

```
.small_block{
padding:5px;
}
```

30. So, our second column looks like this:



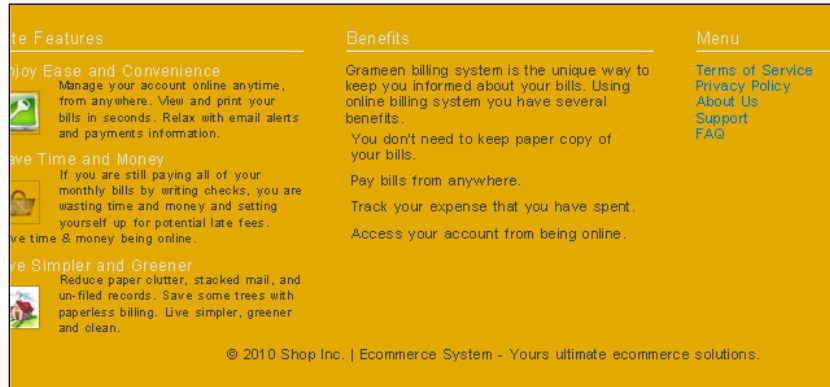
31. Now, we add column three. Following is the HTML code block for the third column:

```
<div class="column column3">
 <div class="footer-box">
 <h3>Menu</h3>

 <a href="#" title="Terms of
Service">Terms of Service
 <a href="#" title="Privacy
Policy">Privacy Policy
 About
Us
 Support
 FAQ

 </div>
</div>
```

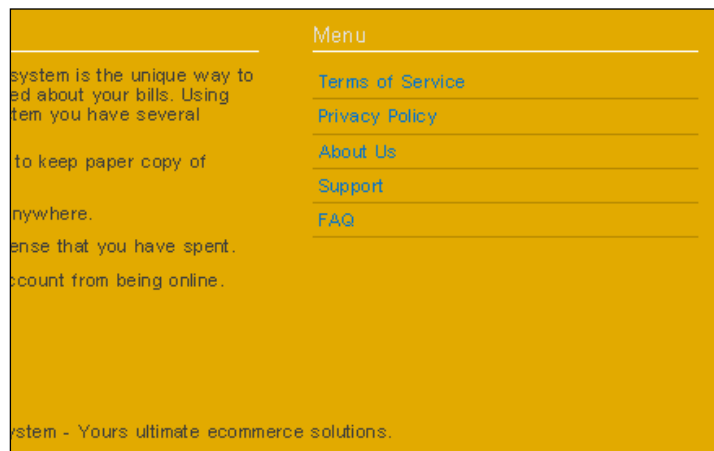
32. When we see the browser, it appears as this:



33. For the styling of the list, we add the following css code:

```
.column ul li{
 color: #333333;
 border-bottom: 1px solid #AF7F00;
 display: block;
 padding: 5px;
}
```

34. And our lists look like the following:



35. We apply a colour to the list's text:

```
.column ul li a{
 color: #333333;
}
```

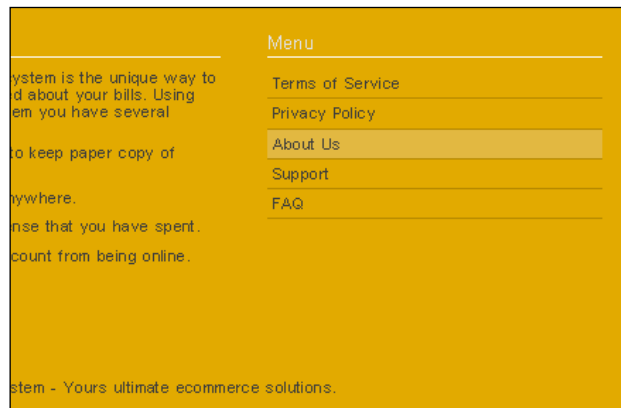
36. Then, the list becomes:



37. When we hover the **li** elements, it will change the background colour of the list.

```
.column ul li:hover{
 background-color: #DFB137;
}
```

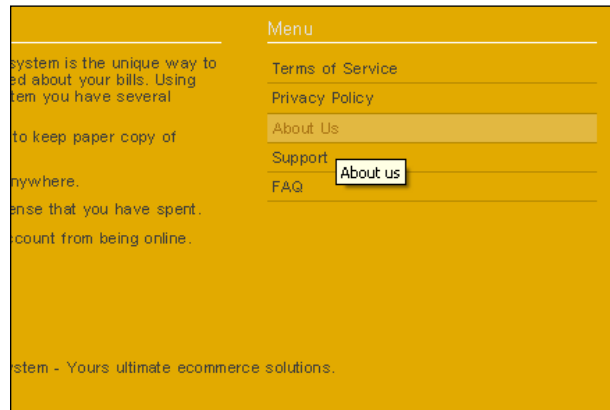
38. This code block brings the following change:



39. On text hover, we change the colour of the text:

```
.column ul li a:hover{
 color: #9f670e;
}
```

40. In the browser, the list looks like this:



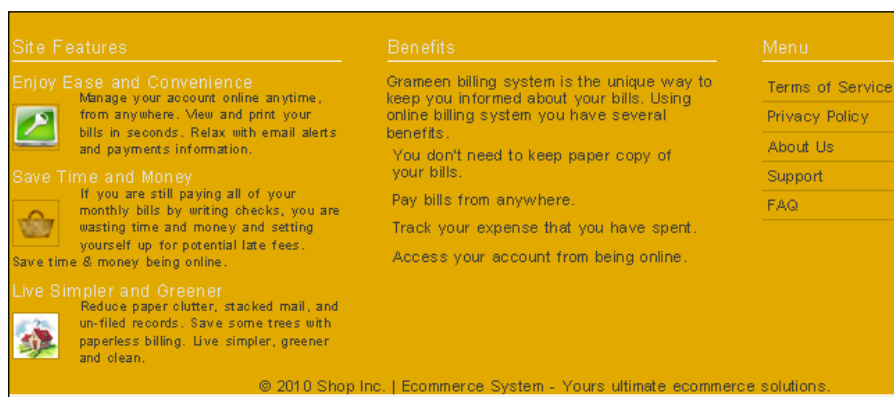
41. At the bottom of the footer, we add the copyright section. The following is the HTML code structure for the copyright block:

```
<div class="clear"></div>
 <div class="copyright">
 <p>© 2010 Shop Inc. | Ecommerce System - Yours
ultimate ecommerce solutions.</p>
 </div>
```

42. For this HTML block, we add the following style code:

```
div.copyright{
margin:0 240px;
}
```

43. Then, our footer becomes like this:







# 9

## Modifying the Administration Panel Theme

In this chapter, we will cover:

- ▶ Modifying headers
- ▶ Creating menus
- ▶ Adjusting the footer
- ▶ Styling the dashboard

### Introduction

In this chapter, you will learn how to color our administration panel. First, we will see how we can modify the header. Then, we will focus on menu creation and move on to change the admin footer. Fourth comes the dashboard styling. We will see a step-by-step guide of this implementation. Let's start our tour!

### Modifying headers

In this recipe, we will modify the header of our store administration. First, we will create the header banner of our admin.

### Getting ready

First, we will open up the `header.tpl` file under `admin\view\template\common`.

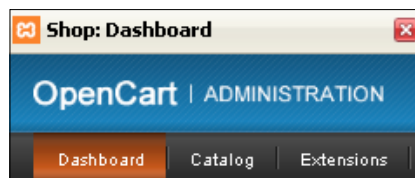
## How to do it...

We follow these steps to modify the header:

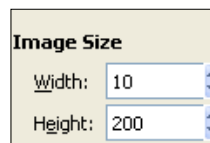
1. In the header file, we customize the title of the admin. We want our store name **Shop** to be displayed in the title bar. We change the title tag and make it the following to display our store name; we can pick the store name from the database also:

```
<title><?php echo 'Shop: '.$title; ?></title>
```

2. Now, our title bar displays our store name:



3. We create a banner image for the header with a dimension of **10x200px**. So, we create an image with GIMP:



4. Choose **#c6a50c** for the foreground color and **#c1c0b9** for the background color:



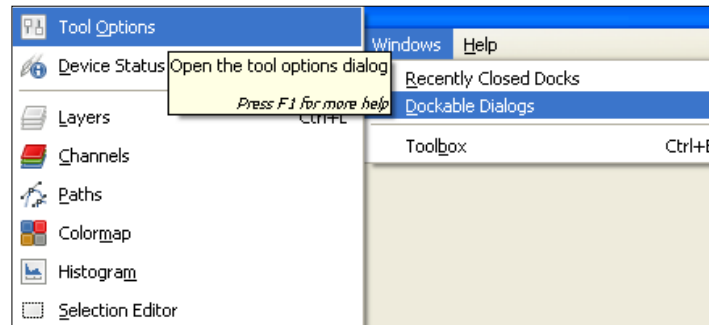
5. Select the **Gradient** tool from the toolbox:



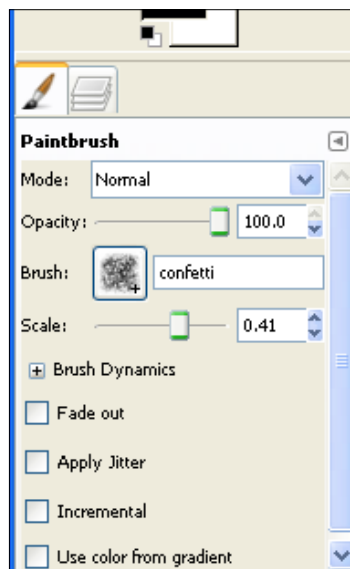
6. Apply some brush effects to our background image. For that we choose the **brush** tool from the tool box:



7. Go to **Windows | Dockable dialogs | Tool options:**



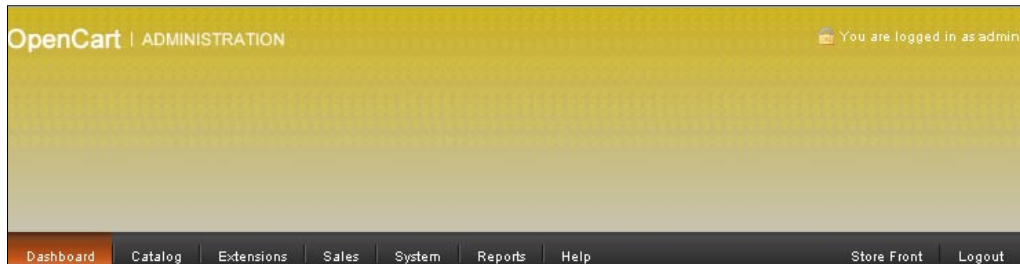
8. Choose the **confetti** brush, scale it to **0.41**. Then, apply it on the background image:



9. We save the image under `admin\view\image`. You can save it as any image file type. Then, we need to change the image path in our stylesheet. We open up the stylesheet for the admin section. It is under `admin\view\stylesheet`. Also, increase the height to 200px.

```
#header {
 background:url("../image/header.png") repeat-x scroll 0 0
 transparent;
 height:200px;
 min-width:900px;
 padding:0 30px;
}
```

10. In the browser, we can view our new header in action:



## How it works...

We have used the background image for the banner of our admin header. You can create different types of banners for our admin in the same way. You can browse the Internet for similar innovative header banners.

## Menu creation

This recipe will show us how we can create a menu for our admin store. We are going to apply some styling to the navigational menu here.

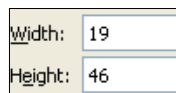
## Getting ready

First, we will open up the `header.tpl` file under `admin\view\template\common`.

## How to do it...

We follow these steps to modify the header:

1. Now, we change the **background** image for our menu. We will create a **19x46px** image with GIMP:



2. And apply **#29a7f6** as foreground color and **#1da2d0** as the background color from the color panel:



3. We save the image under `admin\view\image`.
4. Now, we need to change the image path and the image height in the `stylesheet.css` under `admin\view\stylesheet`:

```
#menu {
 background: url('../image/menu.png') repeat-x;
 position: relative;
 z-index: 1;
 height: 46px;
 clear: both;
 padding: 0px 30px;
 min-width: 900px;
}
```

5. So, in the browser, we find our menu bar as follows:



6. For the selected menu, we create an image of size **19x46px**. We choose the **color picker** tool from tool box. Then, we apply **#07b193** as the foreground color and **#3f9687** as the background color:



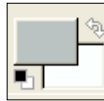
7. We use the **gradient** tool to color our image:



8. We save the image under `admin\view\image`. We can view the change of our menu bar:



9. And the separator image will be **1x46px**. We use **#b4bcbb** as our image background color:



10. We use the **bucket** fill tool from the toolbox to draw the image:



11. You can save the image under `admin\view\image`. So, our menu bar now looks like this:



12. We need to apply some styles in the `stylesheets.css` under `admin\view\stylesheet` to the menu bar texts. We use the following styles:

```
.nav .top, .nav li li.sfhover {
 padding: 13px 15px 9px 17px;
 font-family: Arial, Helvetica, sans-serif;
 font-size: 12px;
 color: #FFFFFF;
 text-align: center;
}
```

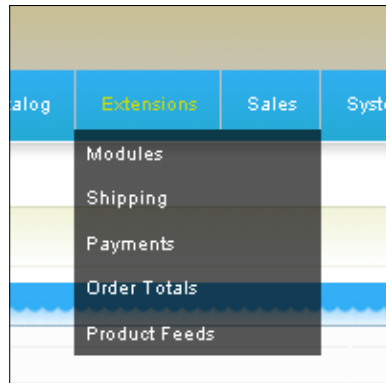
13. So, the text will be placed at the middle:



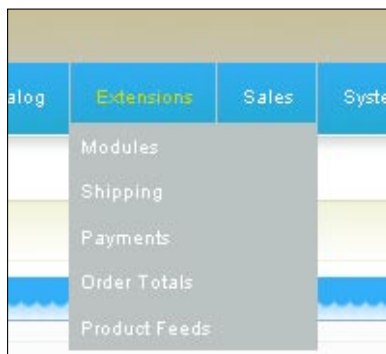
14. We change the hover color of the menu texts:

```
.nav li a:hover {
 color: #c4d509;
}
```

So, the color changes to this:



15. We now create the sub menu background image. We change the color to **#b4bcbb**. We set the dimension at **25x25px**. We use **bucket** fill tool from tool box:





Now, apply a background hover color to the submenu:

```
.nav li li:hover, .nav li li.sfhover {
 background: #29a7f6;
 color: #000000;
}
```

It brings the following changes:



### How it works...

We have used some styling and images for the navigational menu in the header page. You can use your imagination to make your own navigational menu in the header page for the store admin.

## Footer adjustment

This recipe will show us how we can create a footer style for our store admin. We are going to apply some styling to the footer here.

### Getting ready

First, we will open up the `header.tpl` file under `admin\view\template\common`.

### How to do it...

We follow these steps to modify the footer:

1. In the `footer.tpl` file, we placed the following code under the **footer** ID:

```
<div class="column column1">
 <div id="community_news-2">
 <h3 class="widgettitle">Features</h3>
```

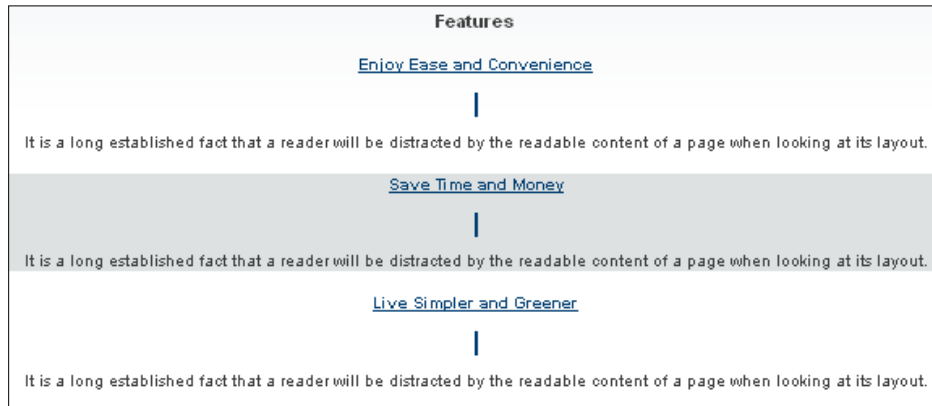
```

<div class="entry box_entry footer-
box">
 <h4><a href="#" rel="bookmark"
title="Permanent Link: Apple rebrands to Banana!">Enjoy Ease
and Convenience</h4>
 <a class="alignleft preloading_
background" href="#" title="Apple rebrands to Banana!">
 <p>It is a long established fact
that a reader will be distracted by the readable content of a
page when looking at its layout.</p>
</div>

<div class="entry box_entry footer-
box">
 <h4><a href="#" rel="bookmark"
title="Permanent Link: Hello world!">Save Time and Money</
h4>
 <a class="alignleft preloading_
background" href="#" title="Hello world!">
 <p>It is a long established fact
that a reader will be distracted by the readable content of a
page when looking at its layout.</p>
</div>
<div class="entry box_entry footer-
box">
 <h4><a href="#" rel="bookmark"
title="Permanent Link: Adobe Wallpaper contest">Live Simpler
and Greener</h4>
 <a class="alignleft preloading_
background" href="#" title="Adobe Wallpaper contest">
 <p>It is a long established fact
that a reader will be distracted by the readable content of a
page when looking at its layout.</p>
</div>
</div>
</div>

```

Our footer then appears as:

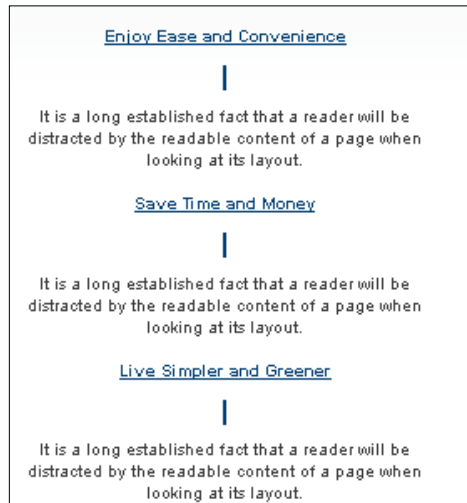


2. We now apply the following style code for our footer column; we make the width fixed:

```
#footer{
color: #333;
position:relative;
width:1000px;
margin:0 auto;
padding-left:0px !important; /*ie6fix*/
padding-left:20px;
z-index:0;
padding-top:40px;
clear: both;
height:350px;
}

.column {
width:293px;
float:left;
padding-left:20px;
margin-right:20px;
}
```

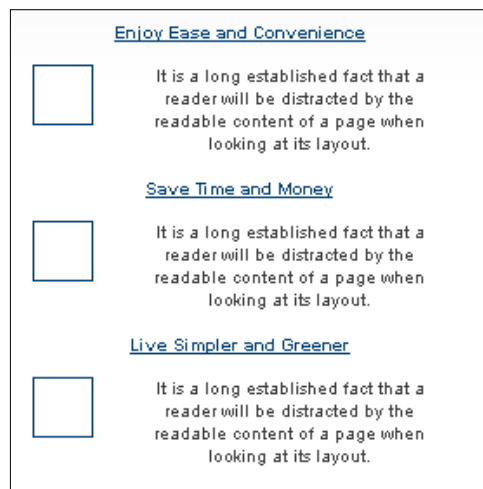
3. See the following footer image:



4. We add the following code for the `alignleft` class:

```
.alignleft{
float:left;
margin:0px 15px 15px 0px;
display: block;
}
```

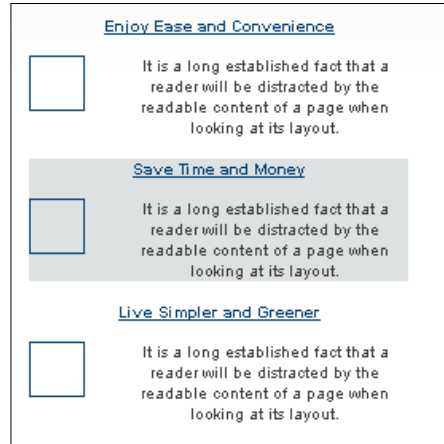
5. Then, our footer column appears as the following:



6. We apply a hover color for the left column blocks:

```
#footer .box_entry:hover {
background-color: #d9dedd;
}
```

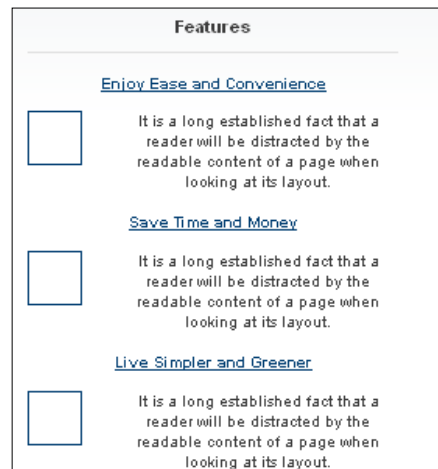
7. When we hover over our blocks, it appears as the following:



8. We add a bottom border and a padding to it. We add a **10px** padding to the top title:

```
#footer h3, #top #footer .box_entry{
border-bottom: 1px solid #ddd;
padding-bottom: 10px;
}
```

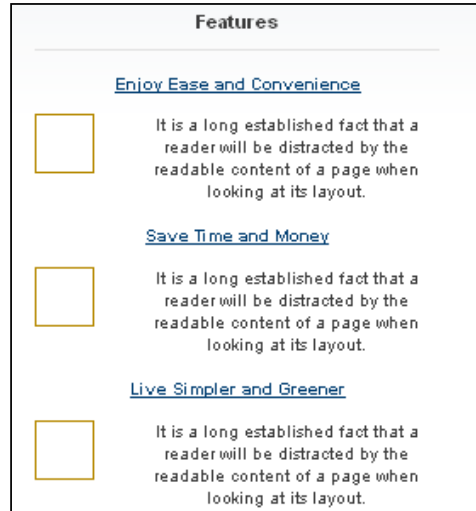
9. This makes the footer title appear as the following:



10. Now change the image border of the block. We set the border color to **#AF7F00**:

```
#footer .footer-box img{
 border:1px solid #AF7F00;
}
```

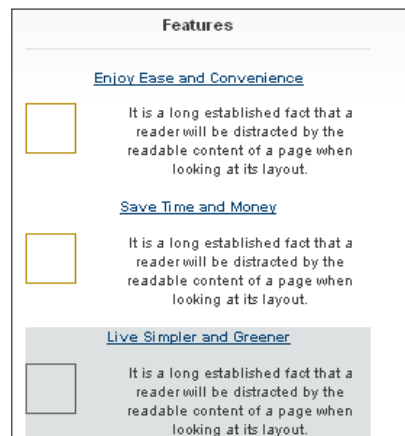
11. So, the image border looks like this:



12. We also apply a code for the image hover style change:

```
#footer .footer-box img:hover{
 border:1px solid #555;
}
```

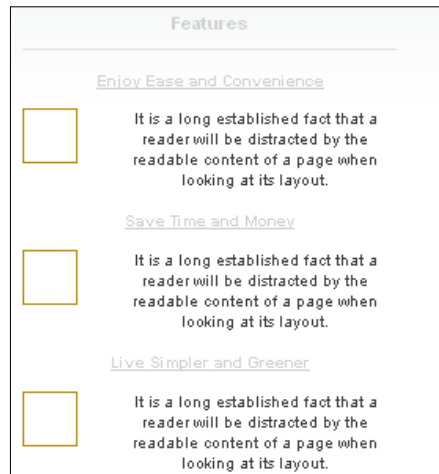
13. When we hover the image area, the border changes to:



14. Now, we change the style for the block titles:

```
#footer h1, #footer h2, #footer h3, #footer h4, #footer h5,
#footer h6, #footer legend, #footer h1 a, #footer h2 a, #footer h3
a, #footer h4 a, #footer h5 a, #footer h6 a{
 color: #ccc;
}
```

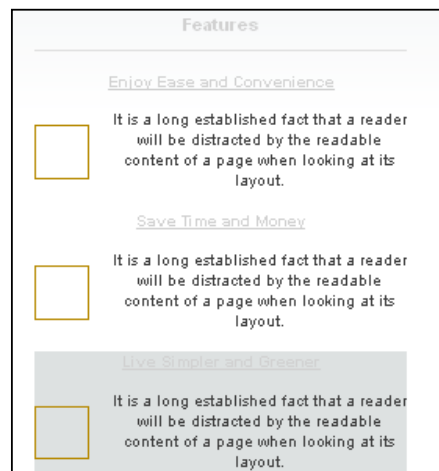
15. Therefore, our footer block titles appear as the following:



16. You now use the following code for hovering on the title:

```
#footer h4 a:hover{
 color: #999;
}
```

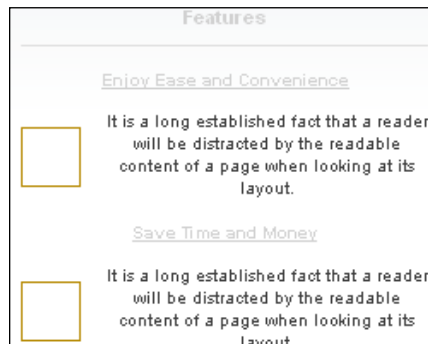
17. Our footer block header title appears as the following:



18. Again, we add the following style into our code block:

```
.box_entry .alignleft{
margin:6px 9px 0px 0;
}
```

19. So, when we look at the first column, it looks like this:



20. Now, it's time to add a new block into our footer. We add the following HTML code block:

```
<div class="column column2">
 <div class="footer-box">
 <h3>Benefits</h3>
 <p>It is a long established fact that a
reader will be distracted by the readable content of a page when
looking at its layout.</p>
 <p class="small_block">You don't need to
keep paper copy of your bills.</p>
 <p class="small_block">Pay bills from
anywhere.</p>
 <p class="small_block">Track your expense
that you have spent.</p>
 <p class="small_block">Access your account
from being online.</p>
 </div>
</div>
```



Our footer contains two blocks and it becomes like the following:

Features	Benefits
<p><u>Enjoy Ease and Convenience</u></p> <p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p>	<p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p> <p>You don't need to keep paper copy of your bills.</p>
<p><u>Save Time and Money</u></p> <p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p>	<p>Pay bills from anywhere.</p> <p>Track your expense that you have spent.</p>
<p><u>Live Simpler and Greener</u></p> <p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p>	<p>Access your account from being online.</p>

21. We use the following styling for the center column:

```
.small_block{
padding:5px;
}
```

So, the list on the center block appears as the following:

Benefits
<p>It is a long established fact that a reader will be distracted by the readable content of a page when looking at its layout.</p> <p>You don't need to keep paper copy of your bills.</p> <p>Pay bills from anywhere.</p> <p>Track your expense that you have spent.</p> <p>Access your account from being online.</p>

And, there will be another block on the footer. Here is the code for that:

```
<div class="column column3">
 <div class="footer-box">
 <h3>Menu</h3>
```

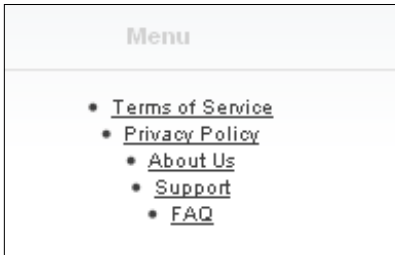
```


 <a href="#" title="Terms of
Service">Terms of Service
 <a href="#" title="Privacy
Policy">Privacy Policy
 About
Us
 <a href="#"
title="Support">Support
 FAQ</
li>

 </div>
</div>

```

22. With the three columns, our footer now looks like this:



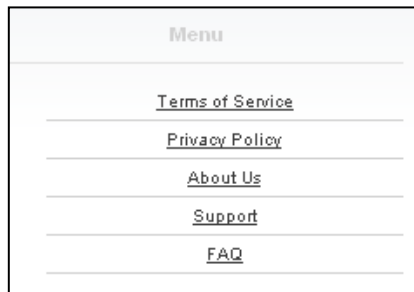
23. For list style, our third block will have the following style:

```

.column ul li{
 color: #333333;
 border-bottom: 1px solid #ccc;
 display: block;
 padding: 5px;
}

```

24. This changes our list:



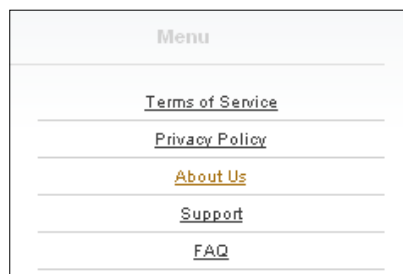
25. We apply a hover color to the list text. But before that we need to remove the following code from the `stylesheets.css` of admin:

```
#footer a {
 color: #333;
 text-decoration: underline;
}
```

Now, we add our new text style code in the style code:

```
.column ul li a{
 color: #333333;
}
.column ul li a:hover{
 color: #9f670e;
}
```

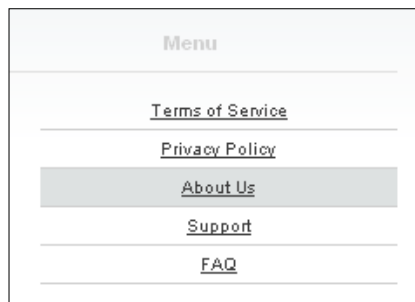
This changes the font color on hover:



26. When we hover the list, the background needs to be changed. For that we add this code block:

```
.column ul li:hover{
 background-color: #d9dedd;
}
```

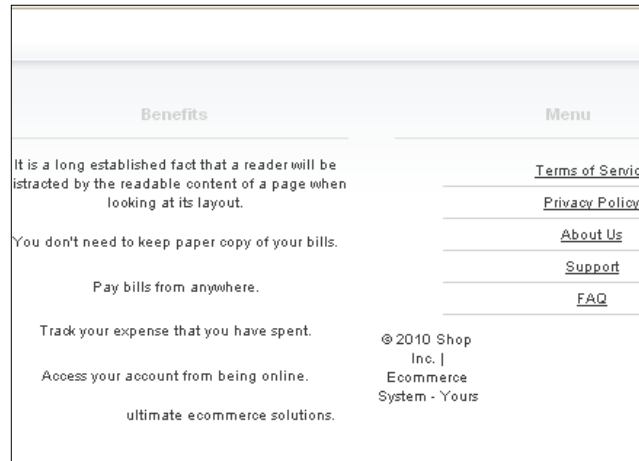
By hovering on the list, the look becomes like this:



27. We add a copyright block in the footer section:

```
<div class="copyright">
 <p>© 2010 Shop Inc. | Ecommerce
 System - Yours ultimate ecommerce solutions.</p>
</div>
```

28. This appears as the following in the browser:



29. We add a clear div before the copyright block:

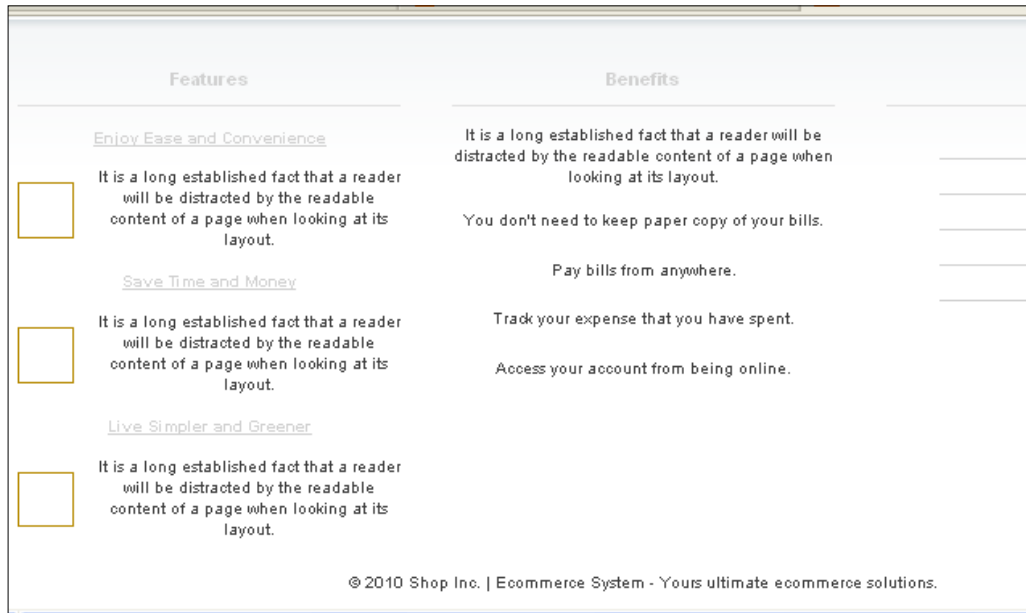
```
<div class="clear"></div>
```

And also use the following styles:

```
#footer{
height: 400px;
}

.column {
height: 100%;
}
```

30. Now, this becomes like the following:



## How it works...

We have used some column divs for the footer page. There are many different types of footer pages. Also, you can use your imagination to make your own footer page.

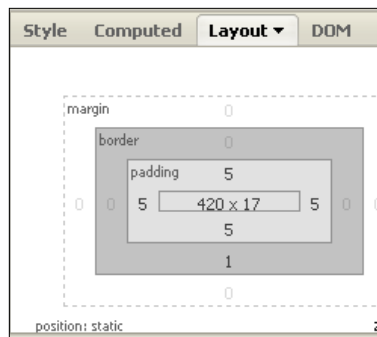
## Styling dashboard

In this recipe, we will style our dashboard. We will apply a header image for each block in the dashboard.

## How to do it...

We follow these steps to change the dashboard:

1. First, we will modify the dashboard header. For that, we first get the dimension of the header from Firebug. We select the **inspector** icon from Firebug.
2. Under the layout section of the right column of Firebug, we found the layout of the header as follows:



3. Now we open up GIMP, and create a new image with 10x27.
4. We zoom our image, so that it will be easy for us to make adjustments. For this, we choose the **zoom** tool:



5. We select the color tool and choose an appropriate color for our dashboard header. For instance, we select **#29a7f6** as the header front end color:



6. Apply the selected color with **gradient** tool from the toolbox. We apply it from top to bottom:



7. Our header image now becomes like this:



8. We also use the **ink tool** from the toolbox:



So, our image becomes like this:



9. We save our image under the `admin\view\image` folder. We need to modify some code in the `home.tpl` file under the `admin\view\template\common` folder. We will replace this code:

```
<div style="display: inline-block; width: 100%; margin-bottom: 15px; clear: both;">
 <div style="float: left; width: 49%;">
 <div style="background: #547C96; color: #FFF; border-bottom: 1px solid #8EAE3; padding: 5px; font-size: 14px; font-weight: bold;"><?php echo $text_overview; ?></div>
 with the following code:
 style="display: inline-block; width: 100%; margin-bottom: 15px; clear: both;">
 <div style="float: left; width: 49%;">
 <div id="dashboard-header"><?php echo $text_overview; ?></div>
div>
```

10. We now add a new selector in our `admin stylesheet.css` under the `admin\view\stylesheet` file. This is the code block in the stylesheet:

```
#dashboard-header {
 background: url('../image/dashboard-header.png') repeat-x;
 color: #FFF;
 border-bottom: 1px solid #8EAE3;
 padding: 5px;
 font-size: 14px;
 font-weight: bold;
}
```

When we refresh our browser, we will find our new dashboard header image:

Overview	
Total Sales:	£0.00
Total Sales This Year:	£0.00
Total Orders:	0
No. Of Customers:	0
Customers Waiting Approval:	0
No. Products:	18
No. Of Reviews:	3
Reviews Waiting Approval:	0

11. Now, we remove the border style of the dashboard header section:

```
#dashboard-header {
border-bottom: 1px solid #8EAECE;
}
```

12. The dashboard header has inline styles. To add this style to other headers, we need to add the selector ID **dashboard-header**. Also, remove the inline styles from the blocks:

```
<div style="background: #547C96; color: #FFF; border-bottom: 1px
solid #8EAECE;">
//...
<div style="background: #547C96; color: #FFF; border-bottom:
1px solid #8EAECE; padding: 5px; font-size: 14px; font-weight:
bold;"><?php echo $text_latest_10_orders; ?></div>
```

13. And we use the following selector in place of the above:

```
<div id="dashboard-header">
```

Our dashboard now looks like this:

Overview	
Total Sales:	£0.00
Total Sales This Year:	£0.00
Total Orders:	0
No. Of Customers:	0
Customers Waiting Approval:	0
No. Products:	18
No. Of Reviews:	3
Reviews Waiting Approval:	0



14. Now, we add some style to the overview section. We apply the **Overview** selector ID to it and use this style:

```
#overview td {
 padding: 2px 0px;
 border-top: 1px solid #eee;
}
```

So, our overview block now looks like this:

Overview	
Total Sales:	£0.00
Total Sales This Year:	£0.00
Total Orders:	0
No. Of Customers:	0
Customers Waiting Approval:	0
No. Products:	18
No. Of Reviews:	3
Reviews Waiting Approval:	0

15. To show the latest orders, we make some coloring on the list. The style is defined in the table **list**. We change the background color to **#e1f1e7**:

```
.list thead td {
 background-color: #e1f1e7;
 padding: 0px 5px;
}
```

This changes the order list style:

Latest 10 Orders			
Order ID	Customer Name	Status	Date Added
No results!			

16. For the border of the list table, we add the following style in the `stylesheets.css`:

```
.list {
 border-collapse: collapse;
 width: 100%;
 border-top: 1px solid #c3e0ce;
 border-left: 1px solid #c3e0ce;
 margin-bottom: 20px;
}
```

```
.list td {
 border-right: 1px solid #c3e0ce;
 border-bottom: 1px solid #c3e0ce;
}
```

So, our order table now appears as the following:

Status	Date Added	Total	Action
No results!			

17. Now, we change the admin section header. For this, we open the `box_center.png` image under `admin\view\image`.

Then, we select the **Select by color** tool from the toolbox:



18. And click on the center area of the selected image. From the color panel, we choose **#f0f3d6**:



19. With the **gradient** tool, we apply it from top to bottom:



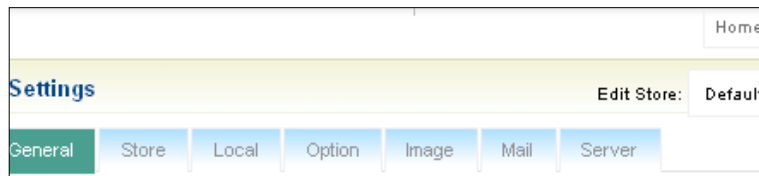
20. We choose **#d1d5ae** for the image top and bottom border:



21. With the **Select by color** tool from the toolbox, we select the top and bottom of the image. And then, we choose the **bucket fill tool**:



22. Then, we save our image and refresh the browser:



23. Now, we change the left side of the block header. We will open the `box_left.png` file under `admin\view\image`.

24. We now select the **Free select** tool from the toolbox:



25. With the selected tool, we select the region of the left white area. And then, we choose **#f0f3d6**. Using the **gradient** tool from the toolbox, we change our selected image and it becomes like this:



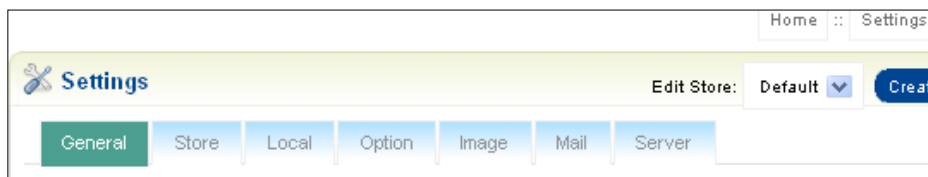
26. We choose the **Fuzzy select** tool from the tool box:



27. And select the border area with it. Using the bucket fill tool, we will apply **#d1d5ae** in the border:



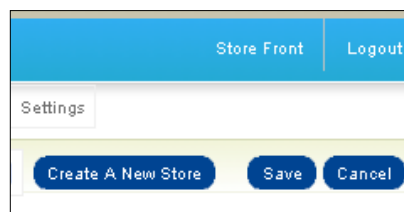
Now, we can view the browser to show the changes:



By following the same above mentioned process, we are going to create the right side image.



So, the right side border now looks like this:



28. We have changed the top header background image. Now, we will adjust the input box style:

```
input[type='text'] {
 padding: 7px;
}
```


This brings the following changes to our input box:

	Image	Product Name ▲
		<input type="text"/>

29. We apply a border color to the input text box:

```
input[type='text'] {
 padding: 7px;
 border: 1px solid #eee;
}
```

Now, our input box is like this:

	Image	Product Name ▲
		<input type="text"/>
<input type="checkbox"/>		Apple Cinema 30"

30. To use a border color on focus, we bring the following style:

```
input[type='text']:focus {
 border: 1px solid #21c3c6;
}
```

By focus on an input text box, the box will appear like this:

Products		
	Image	Product Name ▲
		<input type="text"/>
<input type="checkbox"/>		

31. We have changed the input box style, now we are going to change select box style:

```
select {
 padding: 7px;
}
```

This makes our select box style like this:

Quantity	Status	Action
<input type="text"/>	<input type="text" value="v"/>	<input type="button" value="Filter"/>

32. For the border style, we use this border property:

```
select {
 padding: 7px;
 border: 1px solid #eee;
}
```

33. In the browser, the select box shows up as the following:

Quantity	Status	Action
<input type="text"/>	<input type="text" value="v"/>	<input type="button" value="Filter"/>
100	Enabled	<a href="#">[ Edit ]</a>

34. Now, we will change the **edit** button style in the product section. We wrap the anchor link with the span tag:

```

 <a href="<?php echo $action['href']; ?>">
 <?php echo $action['text']; ?>


```

And add the following style:

```
.edit-button
{
border: 1px solid #eee;
}
```

So, the **Edit** button looks like this:

<a href="#">Edit</a>
----------------------

35. And we apply the `display: block` style to it.

```
.edit-button
{
border: 1px solid #eee;
display: block;
}
```

This makes our style as:



36. Now, we use padding style for the `edit` button:

```
.edit-button
{
border: 1px solid #eee;
display: block;
padding: 5px;
}
```

So, our **Edit** button becomes the following:



37. Then, we align our text on the center:

```
.edit-button
{
border: 1px solid #eee;
display: block;
padding: 5px;
text-align: center;
}
```

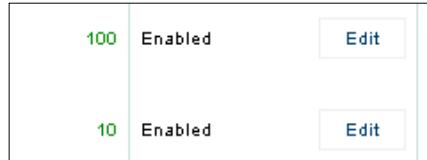
This brings the following changes:



38. We want to remove the underline from the **Edit** button:

```
.edit-button a
{
 text-decoration: none !important;
}
```

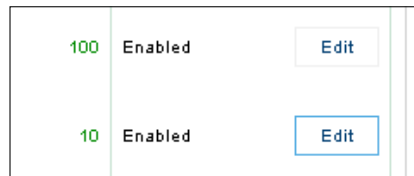
Then, the button appears like this:



39. For hover effect, we write the following code:

```
.edit-button: hover{
border: 1px solid #4ba6e0;
cursor: pointer;
}
```

40. We find the following changes in the browser:

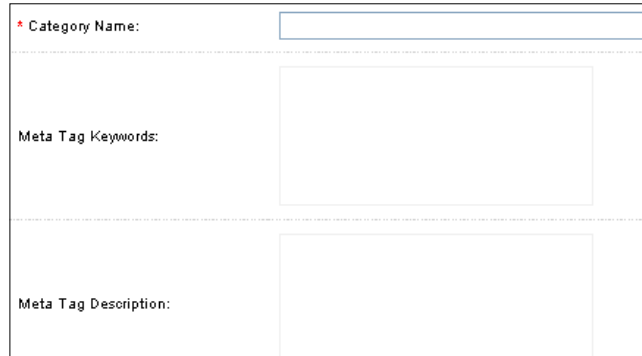


41. We have seen the product list page. Now, let's apply styling to the text area section of the product edit page. We change the text area border color with the following code block:

```
textarea {
 border: 1px solid #eee;
 padding: 7px;
}
```



So, the edit page appears as this:



A screenshot of a web form for editing a category. It contains three main sections: 'Category Name' with a text input field, 'Meta Tag Keywords' with a large text area, and 'Meta Tag Description' with another large text area. Each section has a label on the left and a corresponding input field on the right.

42. Now, we use a focus style to the text area:

```
textarea:focus{
 border: 1px solid #21c3c6;
 padding: 7px;
}
```

We can view the change in the browser:



A screenshot of the same web form as before, but now the 'Meta Tag Description' text area is highlighted with a blue border, indicating it is the active element. The other fields remain unchanged.

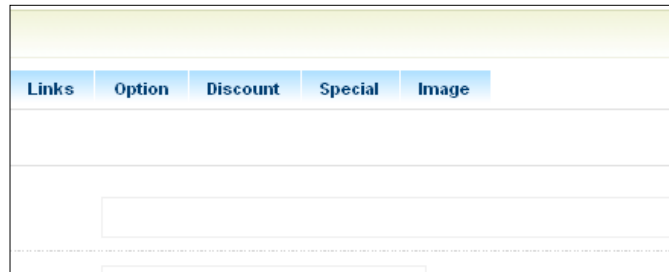
43. We are going to change the settings navigational menu. We first open up `tab.png` under `admin\view\image`. Now, we will change the color to **#a9deff**. We select the color from the color panel:



44. With the gradient tool, we apply it from top to bottom on the image:



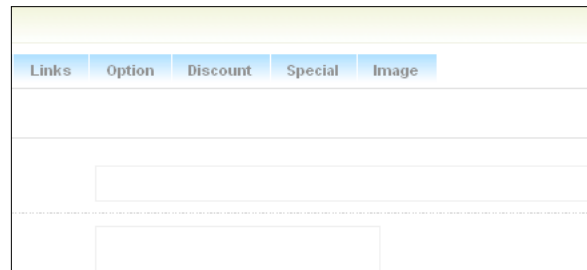
We save the image and go to the browser to view the changes that we have made so far for our settings navigation:



45. Now, change the font color of the navigational tab:

```
.htabs a {
 color: #888;
}
```

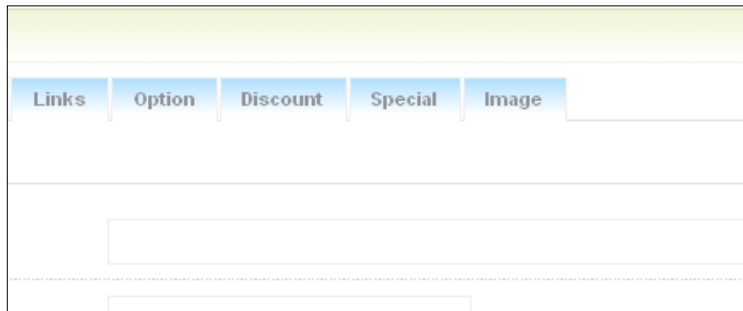
This makes the following changes:



46. We also change the borders of the navigational tabs:

```
.htabs a {
 border-top: 1px solid #eee;
 border-left: 1px solid #eee;
 border-right: 1px solid #eee;
}
```

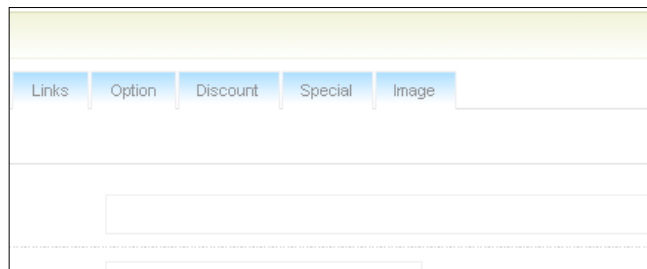
Then, the navigational bar with the new color appears as the following:



47. We change the font weight property to normal for the navigational tab:

```
.htabs a {
 font-weight: normal;
}
```

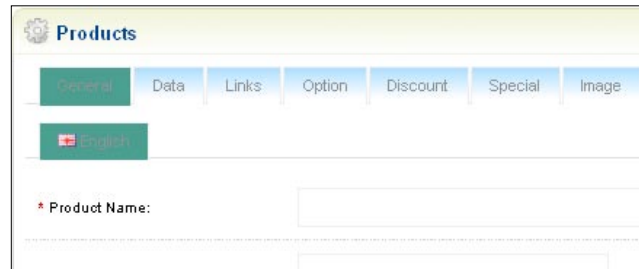
When we refresh the browser, we find our changes in action:



48. For the selected tab, we apply the following styles:

```
.htabs a.selected {
 padding-bottom: 8px;
 background: #3f9687;
}
```

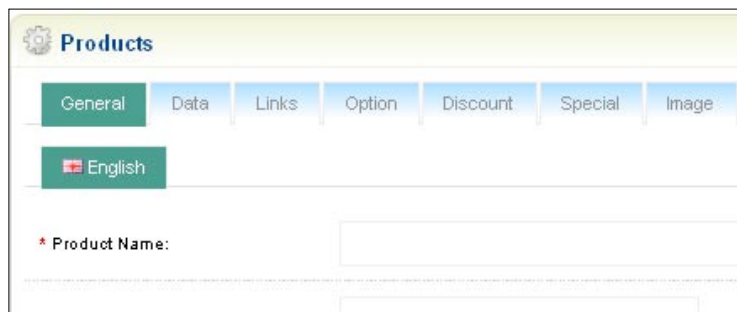
This brings the following changes:



49. Again, we can change the color of the selected tab font:

```
.htabs a.selected {
 padding-bottom: 8px;
 background: #3f9687;
 color: #fff;
}
```

In the browser, we see the new selected tab:



50. To apply hover effect on the navigational tabs, we use a different style. We add the following style:

```
.htabs a:hover {
 background: #a9deff;
}
```

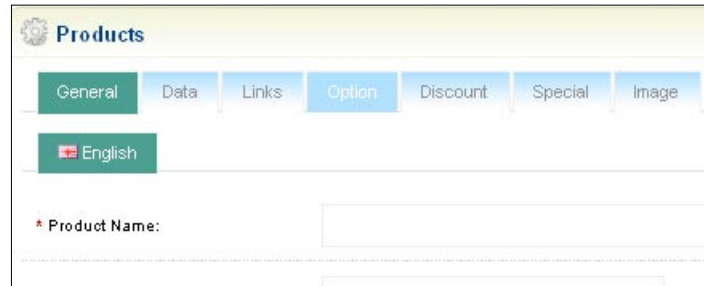
If we go to the browser, we find the changes:



51. To change the font color on hover, use the following code block:

```
.htabs a:hover {
 background: #a9deff;
 color: #fff;
}
```

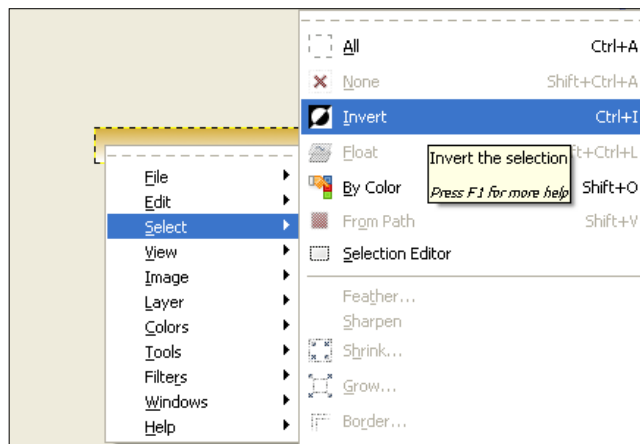
When you visit the browser, it produces the following result:



52. Let's change the color of the **Save** button. First, we open up the `button_right.png` that is under `admin\view\image`. Then, we use the **Select by color** tool from the tool box:



53. We select the transparent area on the image. Now, click the right mouse, go to `select\invert`:



54. We choose **#d2a836** for the button style:



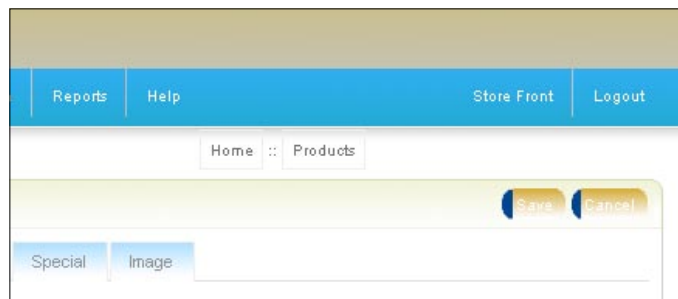
55. We apply the **Gradient** effect on the button with the gradient tool from the toolbox:



56. We apply the effect from the top to bottom with the selected color:



We save our button image and go to the browser to see the changes:

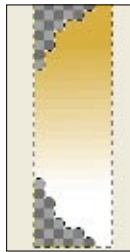


So, we also need to change the `button_left.png` under the `admin\view\image`. We follow a similar approach to style the left button image.

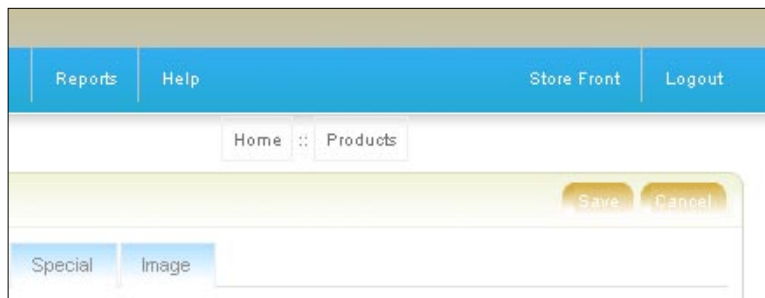
57. First, open it with **Select by color** tool from the tool box:



58. Then, invert the selected region and apply the color with the gradient tool. Our image will become like this:



We save our button and refresh the browser to see the changes in action:



59. The button font color is white; we will change it to #888:

```
.button span {
 color: #888;
 display: block;
 padding: 5px 10px 5px 2px;
 background: url('../image/button_right.png') top right no-repeat;
}
```

After this, the change will be in action:



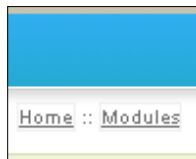
60. Let's change the breadcrumb style for the admin panel. We wrap the anchor link with the class `breadcrumb-link`:

```
<div class="breadcrumb">
 <?php foreach ($breadcrumbs as $breadcrumb) { ?>
 <?php echo $breadcrumb['separator']; ?><span class="breadcrumb-
link"><a href="<?php echo $breadcrumb['href']; ?>"><?php echo
$breadcrumb['text']; ?>
 <?php } ?>
</div>
```

And then, we apply a style on the above class:

```
.breadcrumb-link{
border: 1px solid #eee;
}
```

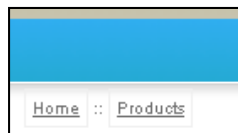
This brings the breadcrumb like this:



61. Now, we apply **5px** padding to the breadcrumbs:

```
.breadcrumb-link{
border: 1px solid #eee;
padding: 5px;
}
```

This makes our header breadcrumbs like the following:

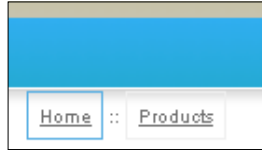


62. We apply a hover color to our breadcrumb header:

```
.breadcrumb-link:hover{
border: 1px solid #4ba6e0;
cursor: pointer;
}
```



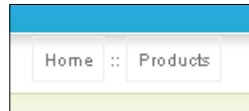
In the browser, we found our breadcrumb as the following:



63. We will remove the underline style from the breadcrumb. For this we apply, the following code:

```
.breadcrumb-link a{
 text-decoration: none !important;
}
```

64. Now, we can find the changes in the browser:



## How it works...

We have used some background images and styles for the different blocks of the admin page. There may be many different blocks of the admin page. Also, you can use your imagination to make your own blocks of admin page.

# 10

## Miscellaneous

In this chapter, we will cover:

- ▶ Creating a 404 page
- ▶ Designing the sitemap page

### Introduction

We have reached the last chapter of our book. So far, we have learnt many different things. We have touched every section of designing an OpenCart store in this tour.

This chapter will provide you information about two common features: the error page and the sitemap. The images describe our each step. So, it's pretty easy to learn the steps. First, we will see how we can create a 404 error page. Then, we will focus on the modification of the sitemap page.

### Broken glass 404 error page

A 404 error indicates that the client was able to communicate with the server, but the server could not find what was requested. In this recipe, we will create a customized 404 error page: a page that appears when a user types a non-existing page of a website. We will place some images along with navigational links on the error page.

### Getting ready

We open the `not_found.tpl` file under the `Catalog\view\theme\shop\template\error`.

## How to do it...

Let's list the steps required to complete the task:

1. First, we will create the HTML structure of the page. We'll replace the existing code of the `not_found.tpl` file with the following code:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://
www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" lang="en"
xml:lang="en">
<head>
<title>Your Store</title>
<meta name="description" content="My Store" />
<link rel="stylesheet" href="catalog/view/theme/shop/stylesheet/
master.css" />
</head>
<body>
//...
</body>
</html>
```

2. In the `master.css` stylesheet, we first reset the browser styles with the following code:

```
@charset "utf-8";

/* Default Values */

a {
 outline: none;
}

div,dl,dt,dd,ul,ol,li,h1,h2,h3,h4,h5,h6,pre,form,fieldset,input,te
xtarea,p,blockquote,th,td {
 margin:0;
}

table {
 border-collapse:collapse;
 border-spacing:0;
}
```

```
fieldset, img {
 border: 0;
}

address, caption, cite, code, dfn, em, strong, th, var {
 font-style: normal;
}

ol, ul {
 list-style: none;
}

caption, th {
 text-align: left;
}

h1, h2, h3, h4, h5, h6 {
 font-size: 100%;
 font-weight: normal;
}

q:before, q:after {
 content: '';
}

abbr, acronym {
 border: 0;
}

object {
 outline: none;
}

q {
 display: block;
}

q:before, q:after {
 content: "";
}
```

3. We have set the reset styles; now, we will apply general styling:

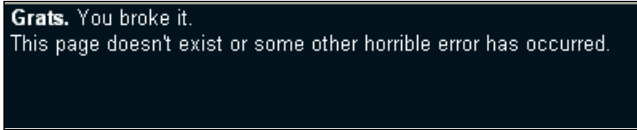
```
/* General layout values */
html {
 height:100%;
}

body {
 height:100%;
 background-color:#010e16;
 margin: 0;
 padding: 0;
 font-family: Arial, Helvetica, sans-serif;
 color:white;
}
```

4. In the body section, we place the following content:

```
<div class="subheader">
 <div class="gratz">
 <p class="lead">Grats. You broke it.</p>
 <p>This page doesn't exist or some other horrible error has
occurred.</p>
 </div>
</div>
```

5. In the browser, we can see our 404 page as this:

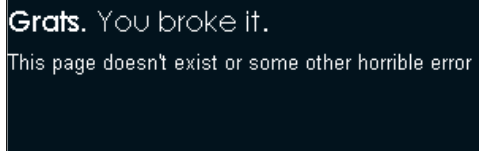


```
Grats. You broke it.
This page doesn't exist or some other horrible error has occurred.
```

6. To change the font style, we are using the following font property:

```
p.lead {
 font:22px/1.964 'Century Gothic', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Verdana, Arial, Helvetica, sans-serif;
}
```

7. This brings the following changes:

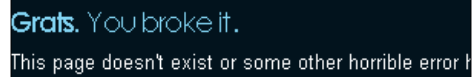


```
Grats. You broke it.
This page doesn't exist or some other horrible error has occurred.
```

8. Let's apply some colour to the texts:

```
p.lead {
 font:22px/1.964 'Century Gothic', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Verdana, Arial, Helvetica, sans-serif;
 letter-spacing:-1px;
 color:#77cfff;
}
```

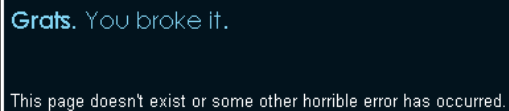
9. Let's look at the browser:



10. Then, apply some margin to it:

```
p.lead {
 font:22px/1.964 'Century Gothic', 'Lucida Sans Unicode', 'Lucida
Grande', 'Lucida Sans', Verdana, Arial, Helvetica, sans-serif;
 letter-spacing:-1px;
 margin:20px 0 1.964em;
 display:block;
 color:#77cfff;
}
```

11. We will go to the browser to find the following changes:



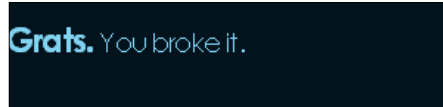
12. To the strong HTML tag, we apply this style:

```
p.lead span {
 display:block;
 padding:.982em 0 0;
}

p.lead strong {
 font-size:28px;
 line-height:1.179;
}
```

```
p.lead sup {
 font-size:.5em;
}
```

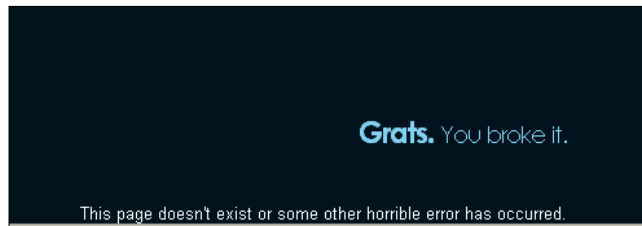
13. Then, the texts appear as the following:



14. In the `gratz` class, we are going to use the following styling for the positioning:

```
.gratz {
 position:absolute;
 top:440px;
 right:180px;
 text-align:right;
 z-index:116;
}
```

15. So, we find the position as this:



16. Again, for the styling of the fonts, we apply the following style on the `gratz` class:

```
.gratz p.lead {
 font-size:36px;
 margin:0 0 5px;
}

.gratz p.lead strong {
 font-size:36px;
}

.gratz p {
 font:12px Arial, Helvetica, sans-serif;
 color:#82B8CB;
 margin:0;
}
```

17. And, in the browser, we find this:



18. We now add some HTML wrapper to the `gratz` class. There is some inline css styling, you can place them in the `master.css` also:

```
<div class="subheader">
 <div class="temp" style="display:block">
 <div class="brokennavcontainer">
 <div class="brokennavloc"
style="background-image:url(catalog/view/theme/shop/image/
brokennav.gif);">

 <div class="brokennav">

 <div class="gratz">
 <p class="lead">Grats. You broke it.</p>
 <p>This page
doesn't exist or some other horrible error has occurred.</p>
 </div>
 </div>
 </div>
 </div>
 </div>
</div>
```

19. Here is the `brokennav.gif`:

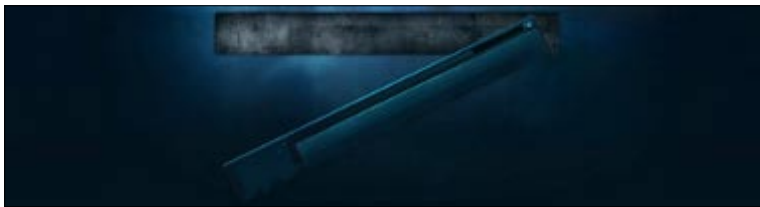




20. Now, we need to use some styling properties on these classes:

```
.brokennavcontainer {
background:url(../image/gratz.jpg) 50% 0 no-repeat;
height:575px;
z-index:110;
margin:0 auto;
position:relative;
text-align:center;
}
```

21. Here is the **gratz** image:



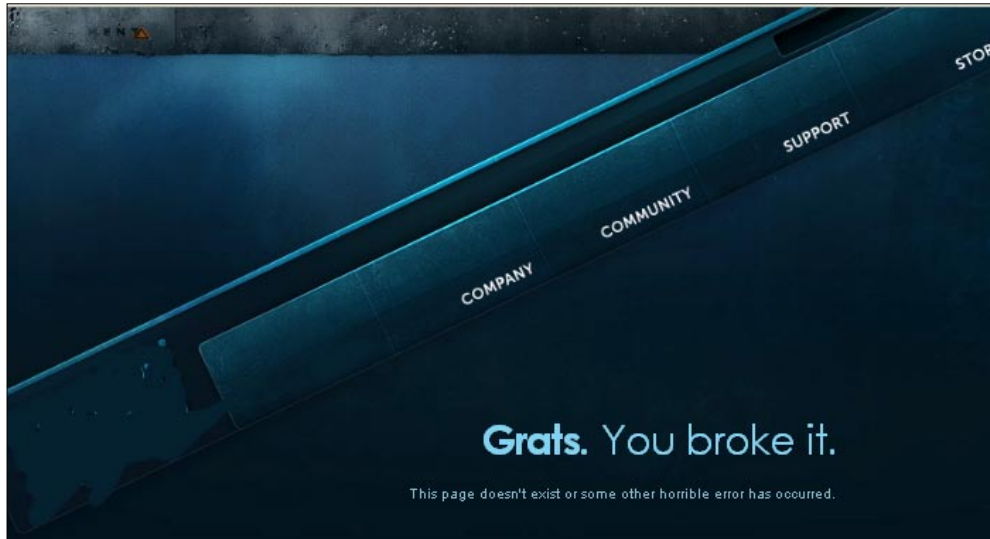
22. We have to use the following extra styles for the above mentioned classes:

```
.brokennavcontainer {
width:954px;
height:572px;
left:0;
position:absolute;
top:0;
left:0;
z-index:117;
}

.brokennavloc {
background-position:50% 0;
background-repeat:no-repeat;
}

.brokennav {
width:954px;
height:552px;
top:0;
z-index:111;
margin:0 auto;
position:relative;
}
```

23. These styles bring the following changes to our 404 page:



24. We add two more classes on the wrapper. So, the code block becomes the following:

```
<div class="subheader">
 <div class="temp" style="display:block">
 <div class="brokennavcontainer">
 <div class="brokennavloc"
style="background-image:url (catalog/view/theme/shop/image/
brokennav.gif);">
 <div class="shatter2"></div>
 <div class="brokennav">

 <div class="gratz">
 <p class="lead">
>Grats. You broke it.</p>
 <p>This page
doesn't exist or some other horrible error has occurred.</p>
 </div>
 </div>
 <div class="shatter1"></div>
 </div>
 </div>
 </div>
 </div>
 </div>
```

25. For the `shatter1` and `shatter2` classes, we will use some images in the styles. Find some broken glass images on the Internet. We will use two broken glass images. First, for the `shatter1` class, we use the following style:

```
.shatter1 {
width:674px;
height:549px;
left:-40px;
z-index:110;
top:0px;
background-image:url(../image/shatter1.png);
position:absolute;
}
```

26. We find the status of our page as this:



27. Then for the `shatter2` class, we also have another image:

```
.shatter2 {
width:262px;
height:285px;
right:0px;
z-index:109;
top:348px;
background-image:url(../image/shatter2.gif);
position:absolute;
}
```

28. With the two images, the page appears like the following:



29. Now, we will add footer after the subheader div on the page. First, we will see the HTML structure of the page:

```
<div class="footercontainer">
 <div class="footer">
 <div class="footercontents">
 <div class="footercontents_int">
 <div class="links">
 |
 Press
 |
 Career Opportunities
 |
 Privacy Policy
 |

 Contact Us
 |

 Sitemap
 </div>

 <p class="dark">
 All trademarks referenced herein
 are the properties of their respective owners.
©2010
 Shops commerce system. All rights reserved.
 </p>
 </div>
 </div>
</div>
```

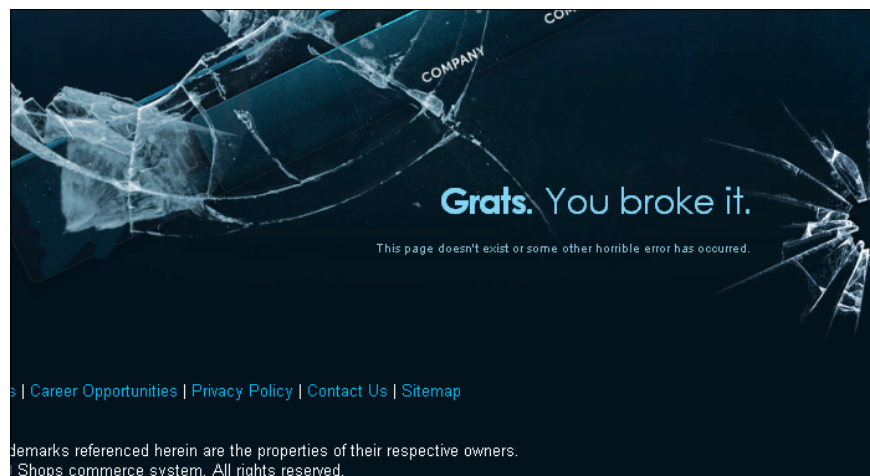
30. With the footer, our 404 page becomes the following:



31. We positioned our footer with the following code block:

```
.footercontainer {
 background-repeat: no-repeat;
 background-position: top center;
 background-color: #010E16;
 width: 100%;
 height: 90px;
 padding: 80px 0 0 0;
}
```

32. This moves the footer to the following position:



33. Now, we will beautify the footer with the following styles:

```
.footer {
 clear: both;
 margin: 0 auto;
 width: 980px;
}

.footer .footercontents {
 width: 100%;
 clear: none;
 height: 60px;
}

.footer .footercontents .footercontents_int {
 color: #FFF;
 font-size: 11px;
 text-align: center;
}
```

34. The above styles make our footer look like this:



35. For the footer navigation, we use the same styling to separate its contents:

```
.footer .footercontents .spacer {
 color: #012839;
 font-size: 13px;
 padding: 0 5px 0 5px;
}
```

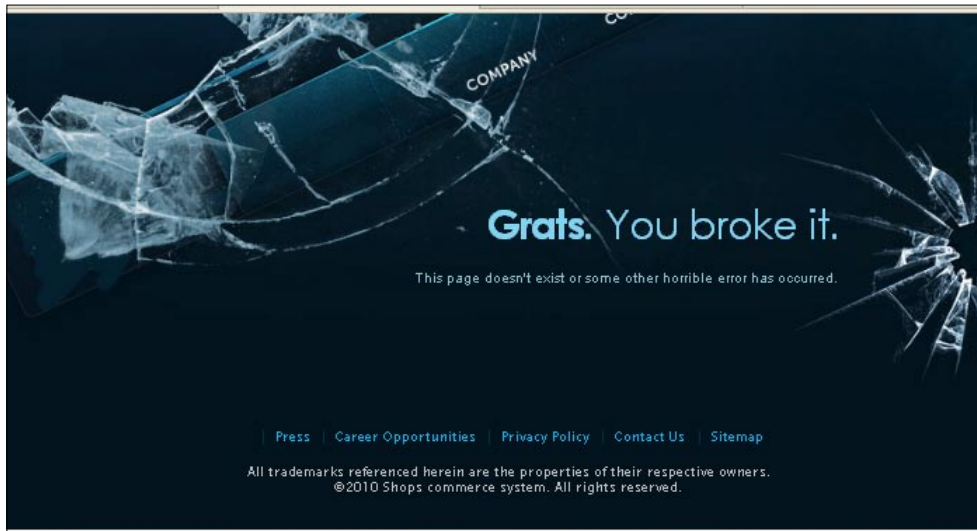
36. The footer navigation changes as the following image shows:



37. To complete our footer styling modification, we add some colour to the copyright texts:

```
.footer .footercontents .links {
 font-family: "Lucida Sans Unicode", "Lucida Grande", sans-
serif;
 line-height: 14px;
 display: inline;
}
.footer .footercontents .dark {
 color: #bbb;
 font-family: "Lucida Sans Unicode", "Lucida Grande", sans-
serif;
 line-height: 14px;
 margin: 0;
}
```

38. Then, this section appears as the following when we refresh our browser:



## How it works...

We have used some background images for the 404 page. We then used z-index for those images. We added styling properties to our page to make it look eye catching.

There are many different designs for the 404 page. Also, you can use your imagination to make your own 404 page.

## See also

- Sitemap creation.

## Sitemap creation

This recipe will show us how we can create a sitemap for our store. Sitemap is the page that has links for the main pages or features of the website. We are going to apply some styling to the sitemap here.

## Getting ready

Let's start our work for sitemap creation. We open up the `sitemap.tpl` file under the `Catalog\view\theme\shop\template\information`.

## How to do it...

Now, we will see a step by step guide on how to modify our sitemap:

1. The default sitemap is created with a two-column table. We will change the structure to **div** tags. And also add a `sitemap-tree` class to the div in the `sitemap.tpl`:

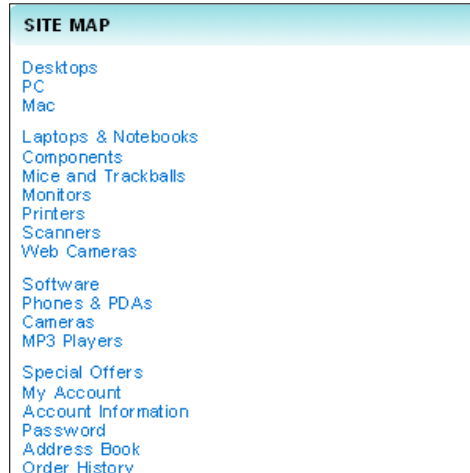
```
<div class="sitemap-tree">
<?php echo $category; ?>
</div>
<div class="sitemap-tree">

 //...

</div>
```



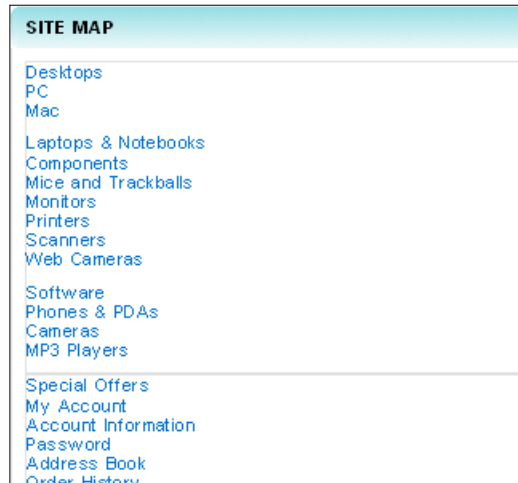
2. With the `div` and without the `table`, our sitemap looks like the following:



3. To the `sitemap-tree` class, we apply the following styles in the `stylesheets.css` under the `Catalog-\view\theme\shop\stylesheet`:

```
.sitemap-tree
{
border: 1px solid #ddd;
}
```

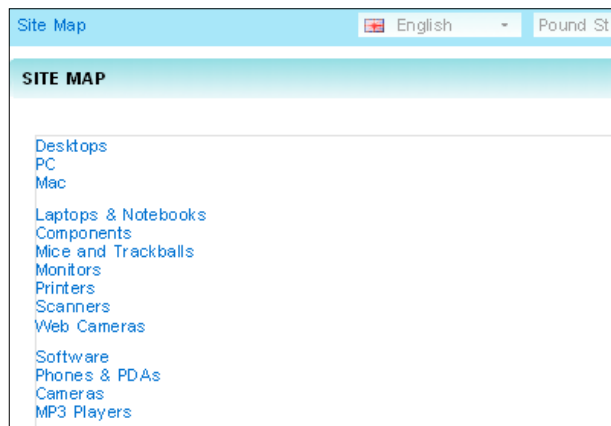
4. With the border, the sitemap appears as the following:



5. And then, we need to apply some margin to separate the borders:

```
.sitemap-tree
{
border: 1px solid #ddd;
margin: 20px 10px;
}
```

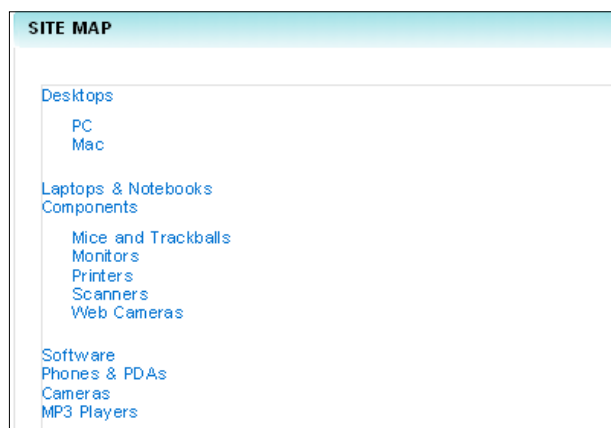
6. This brings a gap between the borders:



7. Now, place some padding to the secondary `ul` tag of our sitemap:

```
.sitemap-tree li ul
{
padding: 10px 25px;
}
```

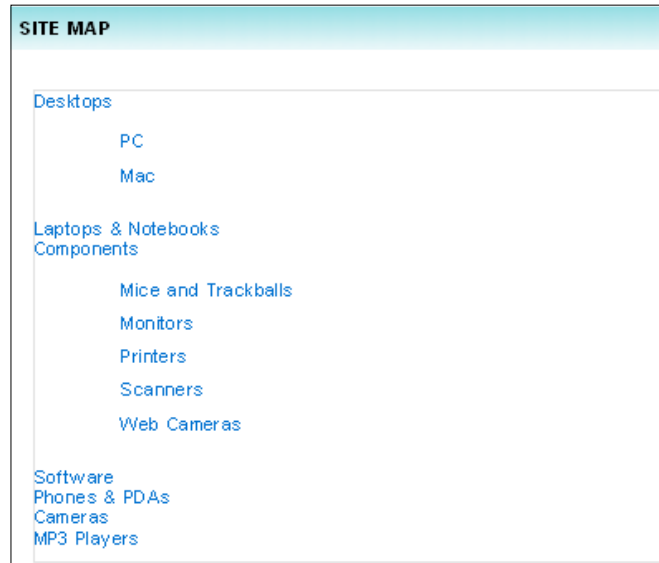
8. This will make the secondary `ul` tag slightly away from the parent `ul` tag:



9. On the child elements of the sitemap, there also needs to be padding:

```
.sitemap-tree ul ul li
{
padding: 5px 40px;
}
```

10. When we go to the browser, it presents the following image:



11. There is some space left on the left side of the child elements. We will place an arrow icon there. Let's create an arrow for it. You can create an arrow icon by yourself. Also, you can search Google for the image. But for now, we will use the following image for the icon:



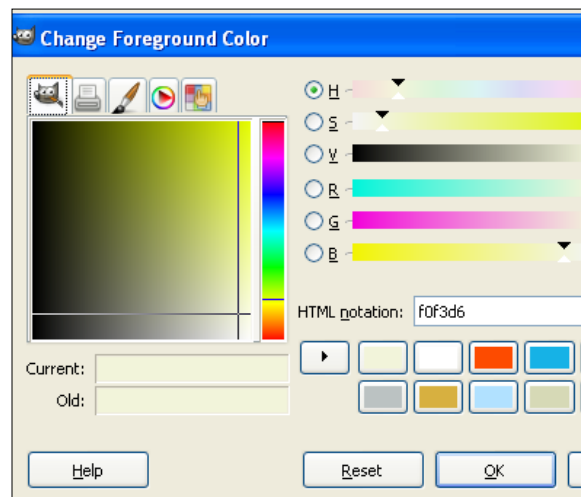
12. We will open up the image with GIMP and then select the **Select by color** tool from the toolbox:



13. We select the centre big arrow with the tool. Now, we will make a copy of it and create a new image with dimensions of **822x502** pixels and choose **Transparent** as the background from the advanced option.
14. Now, we will change the color of the arrow image icon. We select **#c3e0ce** as the arrow color. You can set your favourite one.



15. And also choose **#f0f3d6** as the background image color:



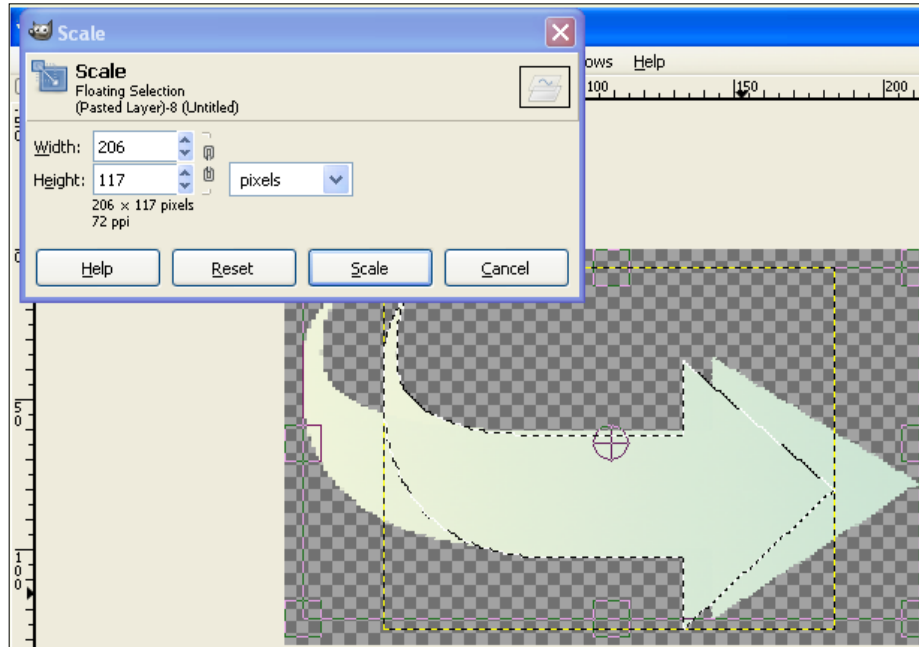
16. We apply this color with the gradient tool from the toolbox:



17. Now, we select the scale tool:

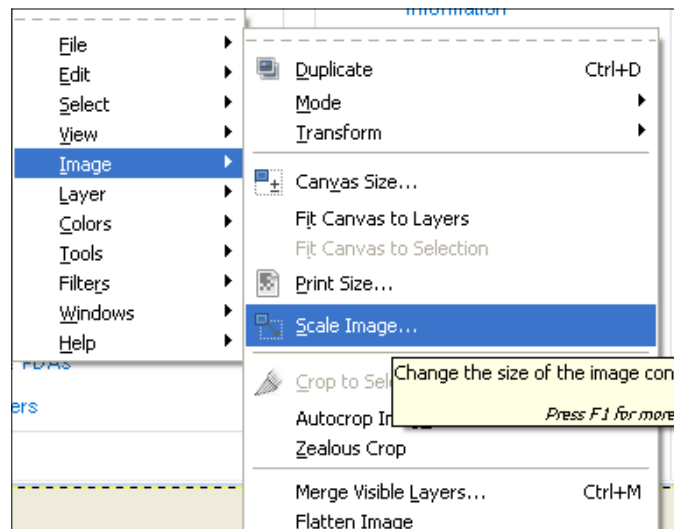


18. We stretch the image to the left and right. So, our image becomes like the following:



19. Then, click on the scale button.

20. Now, we again scale the image by clicking the right button on the mouse:

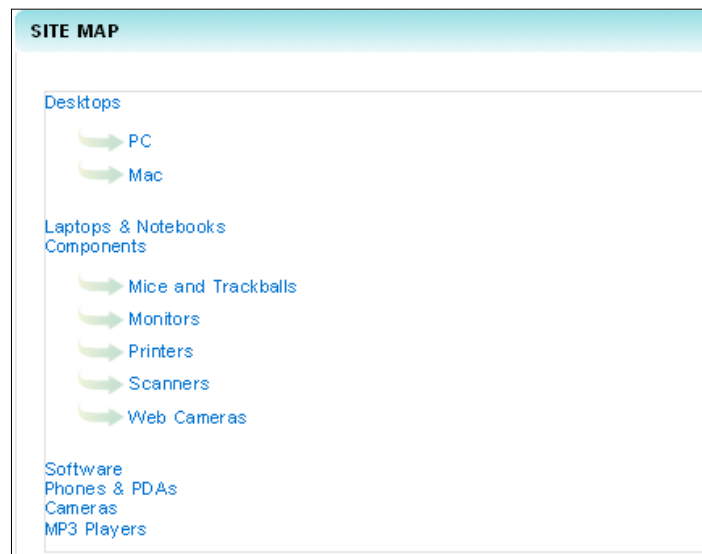


21. We scale it to 22x36 pixels. Then, we save it as `sitemap-arrow.png`.

22. Now, we add the following background style block for it:

```
.sitemap-tree ul ul li
{
background: transparent url('../image/sitemap-arrow.png') no-
repeat;
padding: 5px 40px;
}
```

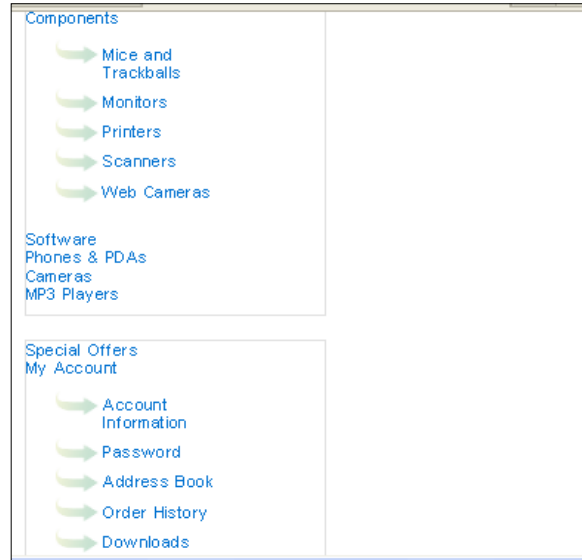
23. This brings the following changes to our sitemap:



24. There are two `div` elements for the sitemap contents. So, we will apply float and a width to `div`.

```
.sitemap-tree{
width:250px;
}
```

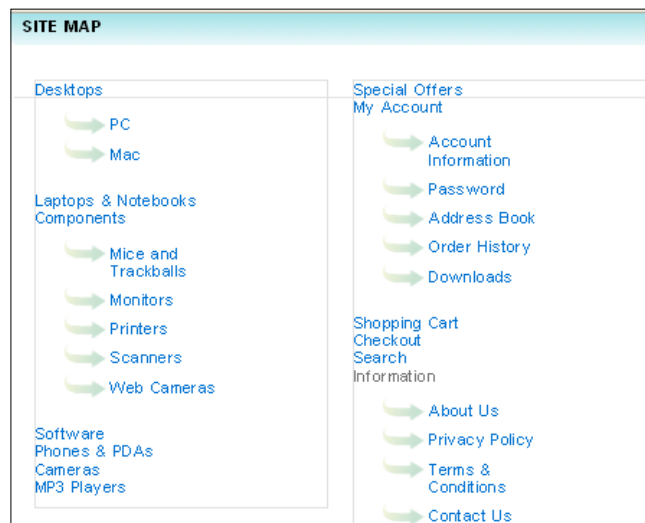
25. Now, our sitemap becomes like this:



26. And then, we apply the float property:

```
.sitemap-tree{
width:250px;
float:left;
}
```

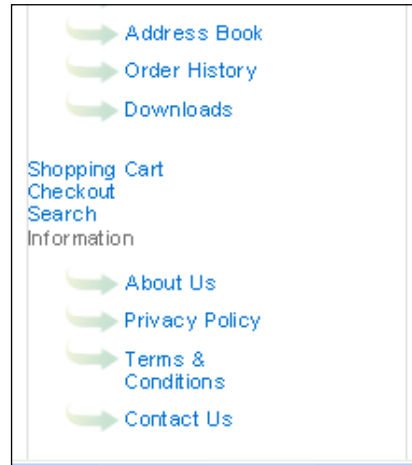
27. This makes the two to be placed by side by side:



28. You will see that the sitemap container's height looks very small now. So, we will use a clear class at the end of the `sitemap-tree` class.

```
</div>
<div class="clear"></div>
<div class="bottom">
```

29. This completes our sitemap for the store **shop**:



### How it works...

We have changed the table-based style to div-based and added an arrow to the links and features. Also, there are many designs of sitemap pages in the Internet. You can see those also.





# Index

## Symbols

**<h1> tag** 27  
**404 error** 281  
**404 error page**  
    creating 282-294  
    font style, modifying 284  
    margin, applying to 285

## A

**admin**  
    header banner, creating for 242, 243  
**admin folder** 33  
**administrator favorites icon**  
    modifying, for OpenCart shop 49-51  
**admin panel**  
    breadcrumb style, modifying for 279, 280  
    entering into 33, 34  
**admin store**  
    footer style, creating for 248-260  
    menu, creating for 244-248  
**Apache 2.2.14** 8

## B

**background menu**  
    adding, to OpenCart store 196-201  
**banner**  
    placing, on site 84, 85  
**basic localization settings**  
    for OpenCart shop 40, 41  
**basic style properties, OpenCart**  
    setting, steps 69-75  
**breadcrumb style**  
    modifying, for admin panel 279, 280

**brokennav.gif** 287  
**brush tool** 242  
**Bucket Fill tool** 181, 266

## C

**category**  
    creating, for products 54-57  
    products, adding to 58-60  
**category block**  
    styles, applying 110-114  
**classes**  
    styling properties, adding to 288-290  
**Colorzilla**  
    about 28  
    download link 28  
    installing 28  
**Colorzilla icon**  
    options 29  
**configuration settings, OpenCart** 11-13  
**cycle plugin**  
    customizing 160-163  
    effect, options 166-174  
    used, for modifying side column 163-166

## D

**dashboard**  
    modifying 260  
**dashboard, styling**  
    border style, removing 263  
    breadcrumb style, modifying for admin panel  
        279, 280  
    browser, viewing for changes 267  
    edit button style, modifying 269

- effects, applying to button image 277, 278
- header, modifying 260
- hover effect, for edit button 271
- image, creating with Gimp 261, 262
- input box style, adjusting 267, 268
- navigational menu settings, modifying 272-276
- padding style, for edit button 270
- select box style, adjusting 269
- style, adding to overview section 264
- working 280

#### **database**

- creating, for OpenCart project 10, 11

#### **designing tools**

- Fireworks 24
- Photoshop 24

#### **downloading**

- GIMP 14

## **E**

#### **edit button style**

- modifying 269

#### **elements**

- inspecting, with Firebug 25

#### **environment**

- setting up, for OpenCart 6

#### **export plugin 23**

## **F**

#### **featured block**

- centre styling 100-109
- placing, at middle 98-100

#### **featured module 89**

#### **features, OpenCart 6**

#### **FileZilla 0.9.33 8**

#### **Firebug**

- about 24
- elements, inspecting with 25
- installing 24, 25

#### **Fireworks 24**

#### **fish-eye menu**

- adding, to OpenCart store 188-195

#### **flip tool 19**

#### **float property 302**

#### **folder structure, OpenCart 61, 62**

#### **font style**

- modifying, for 404 error page 284

#### **footer**

- three column footer, adding 227-239
- wide footer, creating for store 207-215
- with gradient image 216-226

#### **footer style**

- creating, for admin store 248-260

#### **Free select tool 266**

#### **Fuzzy select tool 266**

## **G**

#### **general settings**

- modifying, for OpenCart shop 35-37

#### **GIMP**

- about 14
- default window 15
- downloading 14
- images, creating with 261, 262
- installing 14
- plugins 23

**GNU Image Manipulation Program.** *See* **GIMP**

#### **Google Base 6**

#### **Google Checkout 6**

#### **Google Site Map 6**

#### **gradient image**

- footer with 216-226

#### **gradient tool 261, 265**

#### **gratz class 286**

- HTML wrapper, adding to 287

## **H**

#### **header**

- modifying, of store administration 241-244
- of module, changing 90-93

#### **header.tpl file 241**

#### **header banner**

- creating, for admin 242, 243
- creating, steps 139-142

#### **header structure 118**

#### **horizontal sliding menus**

- adding, to site 176-187

#### **htdocs folder 10**

#### **HTML wrapper**

- adding, to gratz class 287

## I

### IE tester

- about 29
- download link 29

### image properties

- setting, for OpenCart shop 45-49

### images

- creating, for horizontal styling menus 179
- creating, with Gimp 261, 262

### ink tool 262

### input style

- adjusting 267, 268

### inspector icon 260

### installation, Colorzilla 28

### installation, Firebug 24, 25

### installation, GIMP 14

### installation, OpenCart 10

### installation, Web developer 26

### installation, Xampp

- on Windows 8, 9

### installing

- Colorzilla 28
- Firebug 24, 25
- GIMP 14
- OpenCart 10
- Web developer 26
- Xampp 8, 9

## J

### jCarousel

- installing 144, 145
- used, for displaying products 145-153

### jQuery

- installing 144, 145

### jQuery cycle plugin. *See* cycle plugin

## K

### keyword input field 134

## L

### language

- styling, steps 135-137

### language bar

- styling, steps 129-133

### layout

- adjusting 94-98

### layout styles, OpenCart

- resetting 64
- resetting, steps 65-68

### logo

- altering 118
- altering, steps 118, 119
- creating, in OpenCart 15-24

## M

### mail options

- setting 52

### Mamp 9

### master.css stylesheet 282

### menus

- about 122
- creating, for admin store 244-248
- navigational menus 122

### Mercury 4.72 8

### model, view, controller. *See* MVC

### modules

- about 87
- header, changing 90-93
- installing 88

### move tool 19, 183

### MVC 5

### MVC design pattern 62

### MySQL 7

### MySQL 5.1.41 8

## N

### not\_found.tpl file 281, 282

## O

### OpenCart

- about 5
- administration panel 33, 34
- basic style properties, setting 68-75
- configuration settings 11-13
- customizing, cycle plug-in 160-163
- cycle plugin, effects 166-174
- dashboard, modifying 260
- environment, setting up for 6

- featured block, centre styling 100-109
  - featured block, placing at middle 98-100
  - featured module 89
  - features 6
  - folder structure 61, 62
  - footer style, creating for admin store 248-260
  - header banner, creating 138-142
  - headers, modifying of store administration 241-244
  - header structure 118
  - history 5
  - installing 10
  - jCarousel, installing 144, 145
  - jCarousel, used for displaying products 145-152
  - jQuery, installing 144, 145
  - language, styling 135-138
  - language bar, styling 128-133
  - layout, adjusting 94-98
  - layout styles, resetting 64-68
  - logo, altering 118, 119
  - logos, creating in 15-24
  - menu, creating for admin store 244-248
  - menus, placing 122, 124
  - module, installing 88
  - module header, changing 90-93
  - modules 87
  - MVC design pattern 62
  - overview 6
  - payment units, styling 135-137
  - products, displaying vertically 153-159
  - promotional banner, creating 75-84
  - requisites 7
  - search bar, styling 124-128
  - search style, controlling 133-135
  - side column modifying, cycle plugin used 163-166
  - styles, applying to category block 110-114
  - theme 63
  - theme, creating 63, 64
  - theme file style 62
  - top navigation, styling 119, 122
  - OpenCart project**
    - administration area settings 14
    - catalog area settings 14
    - database, creating for 10, 11
    - virtual host, setting up for 11
  - OpenCart shop**
    - administrator favorites icon, modifying for 49, 50, 51
    - basic localization settings 40, 41
    - category, creating for products 54-57
    - general settings, modifying for 35-37
    - image properties, setting for 45-49
    - other options, setting up for 42-44
    - products, adding to category 58-60
    - server options, modifying 53, 54
    - settings, modifying 37-39
    - SMTP mail server, setting up 51
  - OpenCart store**
    - 404 error page, creating for 282-294
    - background menu, adding to 196-201
    - fisheye menu, adding to 188-195
    - sitemap, creating for 295-303
    - social networking style menu, adding to 202-205
  - OpenCart theme**
    - customizing, reasons 6
  - OsCommerce 5**
  - Overview selector ID 264**
- ## P
- payment units**
    - styling, steps 135-137
  - PayPal 6**
  - perl 5.10.1 8**
  - perspective tool 21**
  - Photoshop 24**
  - PHP 5 7**
  - PHP 5.3.1 8**
  - phpMyAdmin 10**
  - phpMyAdmin 3.2.4 8**
  - products**
    - adding, to category 58-60
    - category, creating for 54-57
    - displaying, jCarousel used 145-152
    - displaying, vertically 153-159
  - promotional banner**
    - creating 75
    - creating, steps 75-84

## R

**rectangle select tool** 181

## S

**Sagepay** 6

**scale tool**

about 20, 299

options 20

using 20

**search bar**

styling, steps 124-128

**search style**

controlling, steps 133-135

**select box style**

adjusting 269

**Select by color tool** 265, 266, 277

**server options**

fields 54

modifying 53, 54

**settings**

modifying, of OpenCart shop 37-39

**shatter1 class** 290

**shatter2 class** 290

**shear tool** 21

**shop folder** 33

**side column**

modifying, cycle plugin used 163-166

**Simple Mail Transfer Protocol.** *See* **SMTP mail server**

**site**

banner, placing 84, 85

content section 30

header section 30

horizontal sliding menus, adding to 176-187

latest designing trends 30, 31

running 13

single column structure 30

three column layout 30

two column structure 30

**sitemap**

about 295

creating, for OpenCart store 295-303

**sitemap-tree class** 295

**sitemap.tpl file** 295

**slideOuter class** 177

**SMTP mail server**

mail options, setting 52

setting up 51

**smudge tool**

about 21, 184

options 22, 23

**social networking style menu**

adding, to OpenCart store 202-205

**store administration**

headers, modifying of 241-244

**store shop**

wide footer, creating 207-215

**styles**

applying, to category block 110-114

**styling properties**

adding, to classes 288-290

## T

**templates** 6

**text tool** 182

**theme**

customizing, reasons 6

**theme, OpenCart**

creating, steps 63, 64

**theme file style, OpenCart** 62

**three column footer**

creating 227-239

**top navigation**

styling 119

styling, steps 120, 122

## U

**ul tag** 297

## V

**virtual host**

setting up 11

## W

**Wamp** 9

**Web Developer** 176

**Web developer**

about 26

installing 26

**Web developer toolbar**

tabs 27

**wide footer**

creating, for store 207-215

**Windows**

Xampp, installing on 8, 9

**Worldpay 6****X****Xampp**

about 7

download link 7

environment setup, performing 7-9

installing, on Windows 8, 9

**Xampp 1.7.3**

about 8

contents 8

**Xampp packages 7****x resolution 16****Y****y resolution 16****Z**

z-index 295

Zencart 5

zoom tool 261



## **Thank you for buying OpenCart 1.4 Template Design Cookbook**

### **About Packt Publishing**

Packt, pronounced 'packed', published its first book *"Mastering phpMyAdmin for Effective MySQL Management"* in April 2004 and subsequently continued to specialize in publishing highly focused books on specific technologies and solutions.

Our books and publications share the experiences of your fellow IT professionals in adapting and customizing today's systems, applications, and frameworks. Our solution based books give you the knowledge and power to customize the software and technologies you're using to get the job done. Packt books are more specific and less general than the IT books you have seen in the past. Our unique business model allows us to bring you more focused information, giving you more of what you need to know, and less of what you don't.

Packt is a modern, yet unique publishing company, which focuses on producing quality, cutting-edge books for communities of developers, administrators, and newbies alike. For more information, please visit our website: [www.packtpub.com](http://www.packtpub.com).

### **About Packt Open Source**

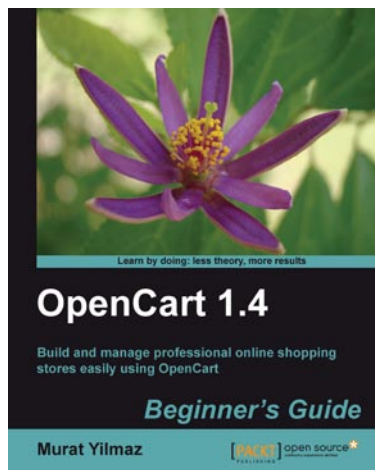
In 2010, Packt launched two new brands, Packt Open Source and Packt Enterprise, in order to continue its focus on specialization. This book is part of the Packt Open Source brand, home to books published on software built around Open Source licences, and offering information to anybody from advanced developers to budding web designers. The Open Source brand also runs Packt's Open Source Royalty Scheme, by which Packt gives a royalty to each Open Source project about whose software a book is sold.

### **Writing for Packt**

We welcome all inquiries from people who are interested in authoring. Book proposals should be sent to [author@packtpub.com](mailto:author@packtpub.com). If your book idea is still at an early stage and you would like to discuss it first before writing a formal book proposal, contact us; one of our commissioning editors will get in touch with you.

We're not just looking for published authors; if you have strong technical skills but no writing experience, our experienced editors can help you develop a writing career, or simply get some additional reward for your expertise.





## OpenCart 1.4 Beginner's Guide

ISBN: 978-1-849513-02-9

Paperback: 240 pages

Build and manage professional online shopping stores easily using OpenCart

1. Develop a professional, easy-to-use, attractive online store and shopping cart solution using OpenCart that meets today's modern e-commerce standards
2. Easily integrate your online store with one of the more popular payment gateways like PayPal and shipping methods such as UPS and USPS
3. Provide coupon codes, discounts, and wholesale options for your customers to increase demand on your online store
4. With hands-on examples, step-by-step explanations, and illustrations



## Joomla! Template Design

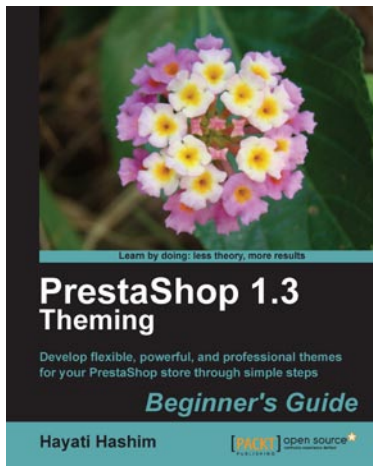
ISBN: 978-1-847191-44-1

Paperback: 232 pages

A complete guide for web designers to all aspects of designing unique website templates for the free Joomla! 1.0.8 PHP Content Management

1. Create Joomla! Templates for your sites
2. Debug, validate, and package your templates
3. Tips for tweaking existing templates

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles

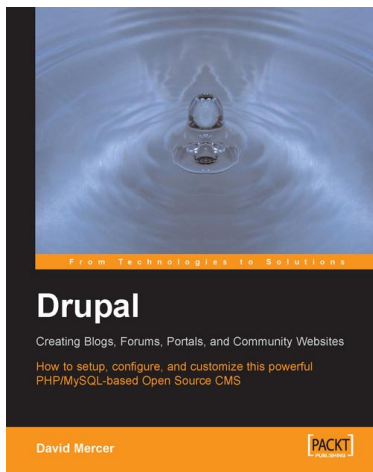


## **PrestaShop 1.3 Theming – Beginner's Guide**

ISBN: 978-1-849511-72-8      Paperback: 312 pages

Develop flexible, powerful, and professional themes for your PrestaShop store through simple steps

1. Control the look and feel of your PrestaShop store by creating customized themes
2. Learn the tips and tricks to make theming in PrestaShop easier
3. Create your own PrestaShop theme in a few simple steps
4. Learn how to use PrestaShop back office panel to make simple changes to your theme as well as tweaking the right lines of code to customize the look of your store



## **Drupal**

ISBN: 978-1-904811-80-0      Paperback: 284 pages

How to setup, configure and customise this powerful PHP/MySQL based Open Source CMS

1. Install, configure, administer, maintain and extend Drupal.
2. Control access with users, roles and permissions
3. Structure your content using Drupal's powerful CMS features
4. Includes coverage of release 4.7

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles



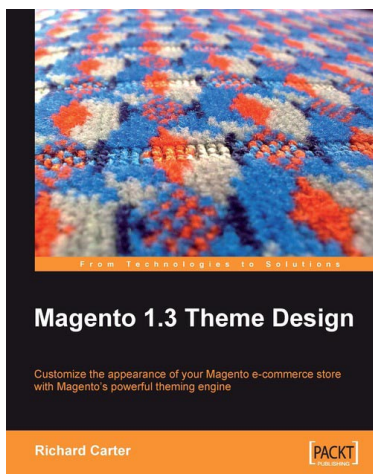
## Joomla! 1.5 Template Design

ISBN: 978-1-847197-16-0

Paperback: 284 pages

Create your own professional-quality templates with this fast, friendly guide

1. Create Joomla! 1.5 Templates for your sites
2. Debug, validate, and package your templates
3. Tips for tweaking existing templates with Flash, extensions and JavaScript libraries



## Magento 1.3 Theme Design

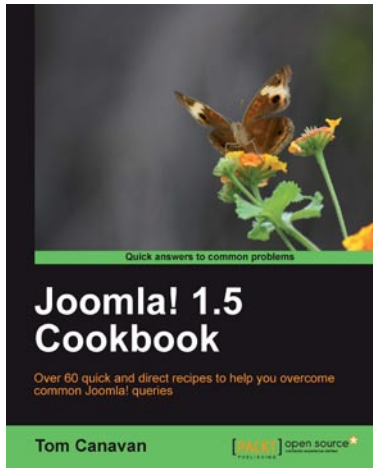
ISBN: 978-1-847196-64-4

Paperback: 188 pages

Customize the appearance of your Magento e-commerce store with Magento's powerful theming engine

1. Give your Magento stores a unique branded look and feel by creating your own Magento themes
2. Use design techniques to reinforce your brand message and increase sales
3. Customise your Magento theme's look, feel, layout, and features
4. Promote and improve your Magento store with the use of social media such as Twitter, social bookmarks, and so on
5. Ideas and examples for defining a good print stylesheet

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles

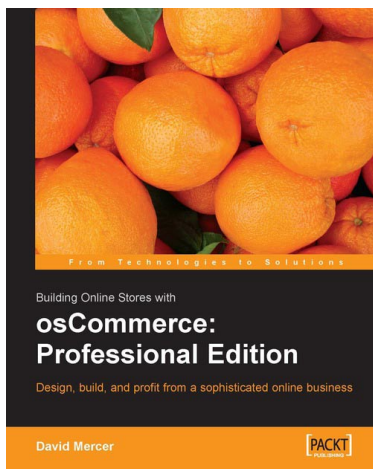


## Joomla! 1.5 Cookbook

ISBN: 978-1-84951-236-7      Paperback: 340 pages

Over 60 quick and direct recipes to help you overcome common Joomla! queries

1. Find quick solutions to common Joomla! problems
2. Part of Packt's Cookbook series: Each recipe is a carefully organized sequence of instructions to complete the task as efficiently as possible
3. Look at recipes that cover the portions of Joomla! 1.6 that are brand new
4. Over 60 practical recipes covering a range of site management and core Joomla! activities



## Building Online Stores with osCommerce: Professional Edition

ISBN: 978-1-904811-14-5      Paperback: 388 pages

Learn how to design, build, and profit from a sophisticated online business

1. Install, configure, and customize osCommerce
2. Enhance and modify osCommerce
3. Learn from a sample, fully functional site packed with useful features such as gift certificates and discounts, cross- and up-selling, RSS feed aggregation, enhanced product image handling and bug fixes

Please check [www.PacktPub.com](http://www.PacktPub.com) for information on our titles