[PACKT] open source
community experience distilled
PUBLISHING

OpenCart 1.4 Template Design Cookbook

Tahsin Hasan

Chapter No. 3
"Layout Structure"

In this package, you will find:

A Biography of the author of the book

A preview chapter from the book, Chapter NO.3 "Layout Structure"

A synopsis of the book's content

Information on where to buy this book

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.

OpenCart 1.4 Template Design Cookbook

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 willl 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.

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.

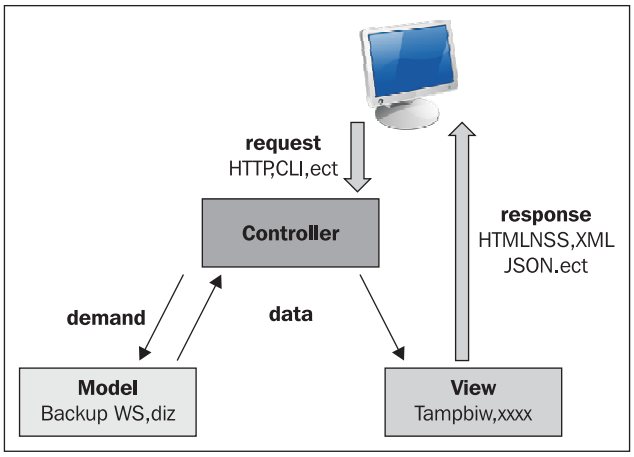
Then, we'll do some basic styling for the site. For each change in style, we will see why we did that.

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

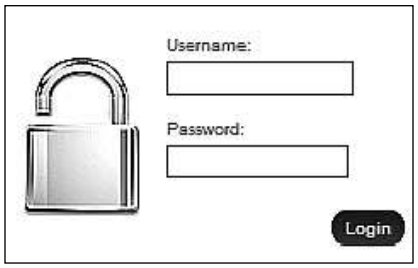
Following are the steps for creating a new theme for OpenCart:

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:

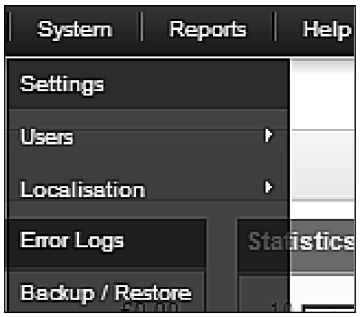
```
      <link rel="stylesheet" type="text/css"
         href="catalog/view/theme/shop/stylesheet/stylesheet.css"
/>
       // other lines ...
       <link rel="stylesheet" type="text/css"
         href="catalog/view/theme/shop/stylesheet/ie6.css" />
      //other lines ...
       <div class="div3">
       <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;
}
```

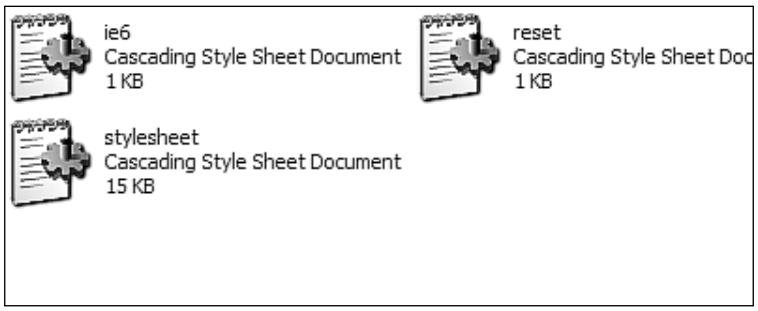
```
blockquote:before, blockquote:after,
q:before, q:after {
   content: '';
   content: none;
}

: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.

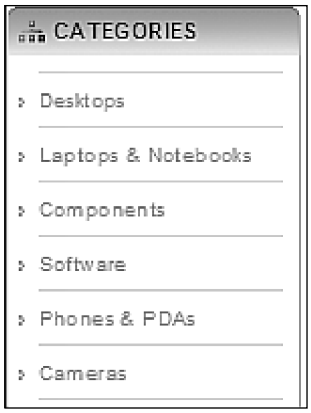
How to do it

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.

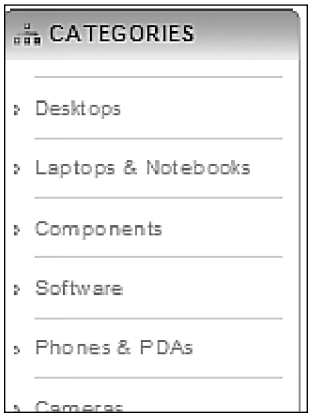


3. Here, the letters are moved slightly from each other.

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;
}
```

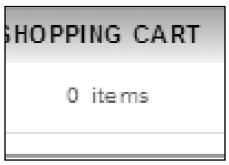
10. Here, we set the font size of some heading styling properties. We can adjust the font size according to our need. We set it for the overall site. But, it will not affect the styling of the main container as there is also a styling property available under the `content` ID. We will see how to change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



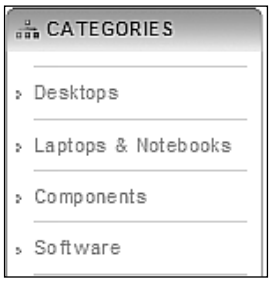
14. We also apply our font size for the site. So, the code block changes to this:

```
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.

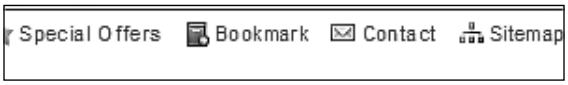
```
a, a:visited {
   color: #1B57A3;
   text-decoration: underline;
   cursor: pointer;
}
```

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:

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.

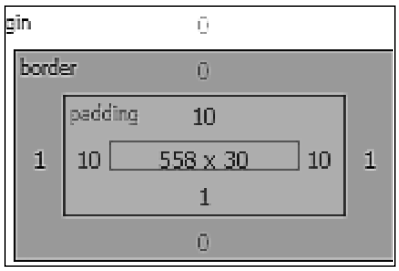
How to do it

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:

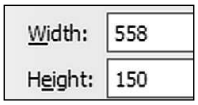
2. It will open up Firebug; now click on the firebug **inspector** icon:

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:

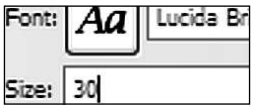
10. Now, pour your selected colour on the banner image area. I have chosen the following color (maroon):

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**:

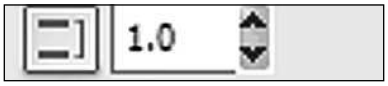


15. Choose the font colour #ffffff.

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

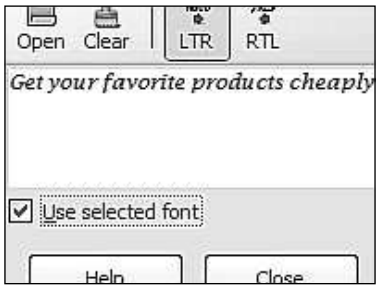
17. We leave indentation to **0px**:

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:

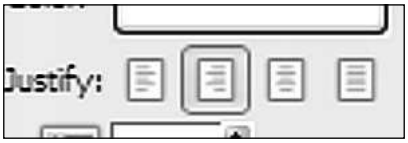
23. Now, we write a subtext below the above text in our banner. We will use a different font style to write this. But you can use the same style if you wish.
24. You can see currently our text selection area covers most of the right side of the image. To write a subtext, we need to shorten the height 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**:

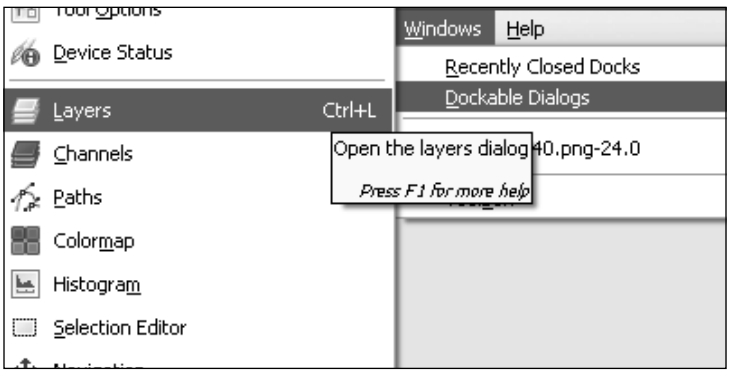
28. We need to change the font size also. We will change it to **20px**:

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

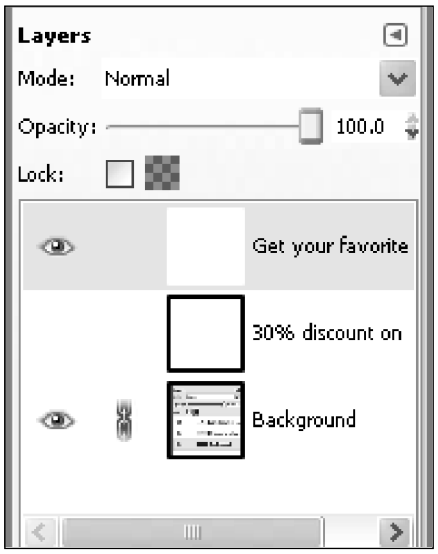


30. Now, select an area below the current text on the banner to write subtext. Our banner image will become like the following:

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



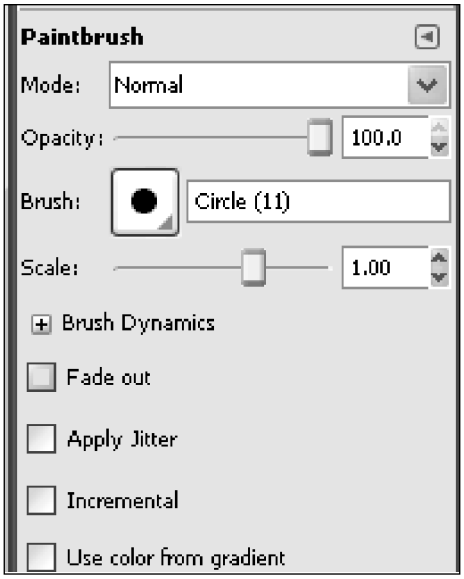
32. We will see that the following layers window opens:



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:

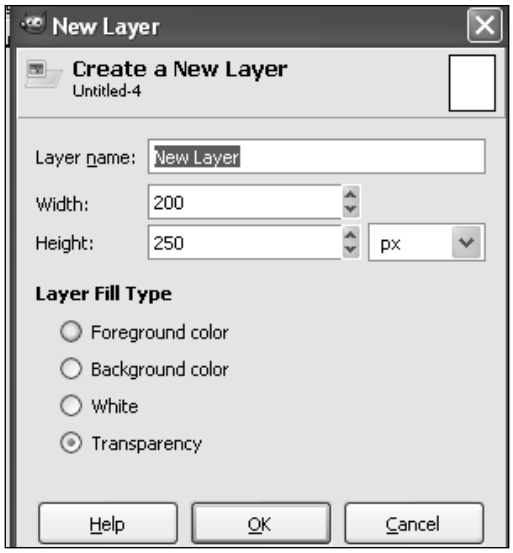
38. Under the brush dynamics, you can see many different options. You can use them according to your need. We choose the following gradient for our brush:

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.

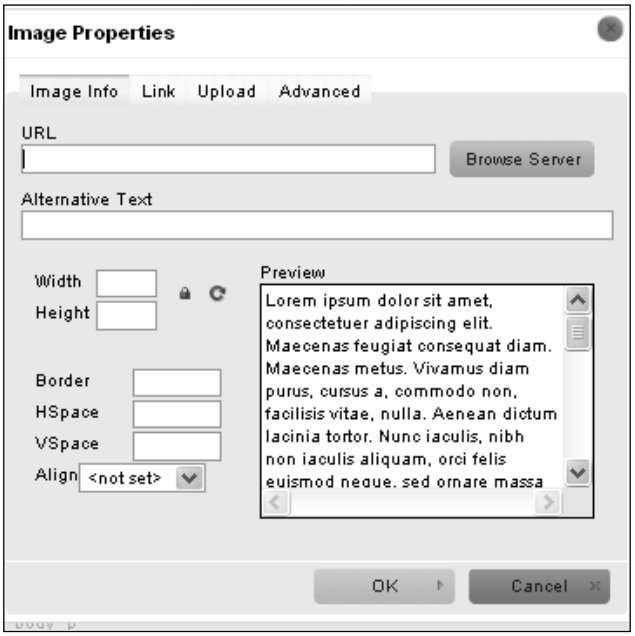
Getting started

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.

Where to buy this book

You can buy OpenCart 1.4 Template Design Cookbook from the Packt Publishing website: `https://www.packtpub.com/opencart-1-4-template-design-cookbook/book`

Free shipping to the US, UK, Europe and selected Asian countries. For more information, please read our shipping policy.

Alternatively, you can buy the book from Amazon, BN.com, Computer Manuals and most internet book retailers.

www.PacktPub.com

For More Information:
www.packtpub.com/opencart-1-4-template-design-cookbook/book