

# RapidWeaver 3.6 Theme Tutorial

The aim of this tutorial is to give you a basic understanding of how you can build a unique theme for RapidWeaver. You'll learn about theme syntax, the property list file and how it controls your theme, how themes are structured and some basic styling.

**Requirements:** RapidWeaver 3.6 or newer, TextWrangler (or similar), CSSEdit (or similar)

**Skill Level:** RapidWeaver / Web enthusiast. Basic XHTML and CSS knowledge an advantage.

**Objection:** To gain an understanding of how RapidWeaver themes are constructed and learn some basic CSS styling for the theme.

**Time:** 1 - 2 hours depending on skill level.

## *Before you start*

Before you start you need to make sure you have RapidWeaver 3.6 or newer installed. You can download the latest version of RapidWeaver from <http://www.realmacsoftware.com>

You'll also need a text editor, TextWrangler from Bare Bones Software (<http://www.barebones.com>) is free. Alternatively you can use TextEdit.

Finally a good CSS editor will also come in very useful. I recommend you buy a license for CSSEdit from MacRabbit (<http://macrabbit.com>), it's not essential but it makes CSS coding a breeze!

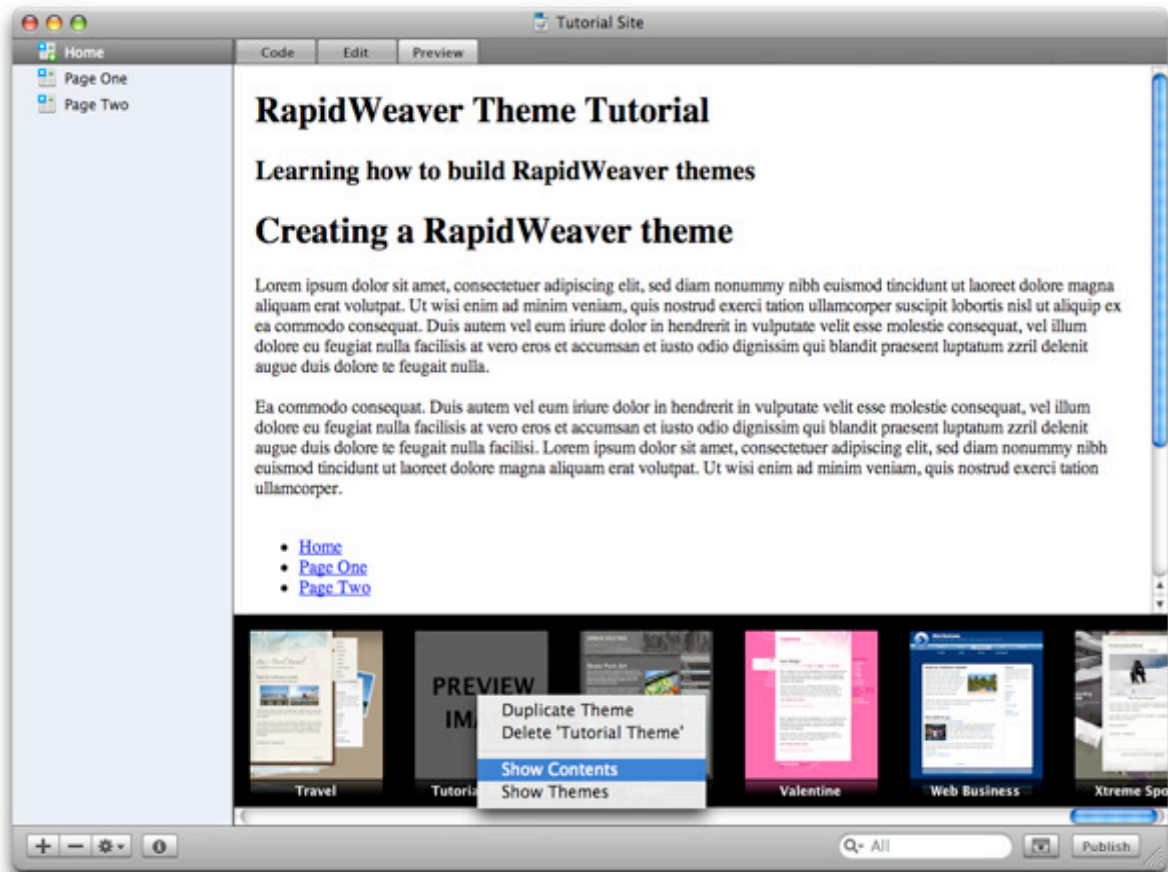
With everything in place, let's get started.

## Installing Your Theme

Install the *Tutorial.rwtheme* file by double-clicking it. You then need to restart RapidWeaver for the theme to become active.

### Access The Files

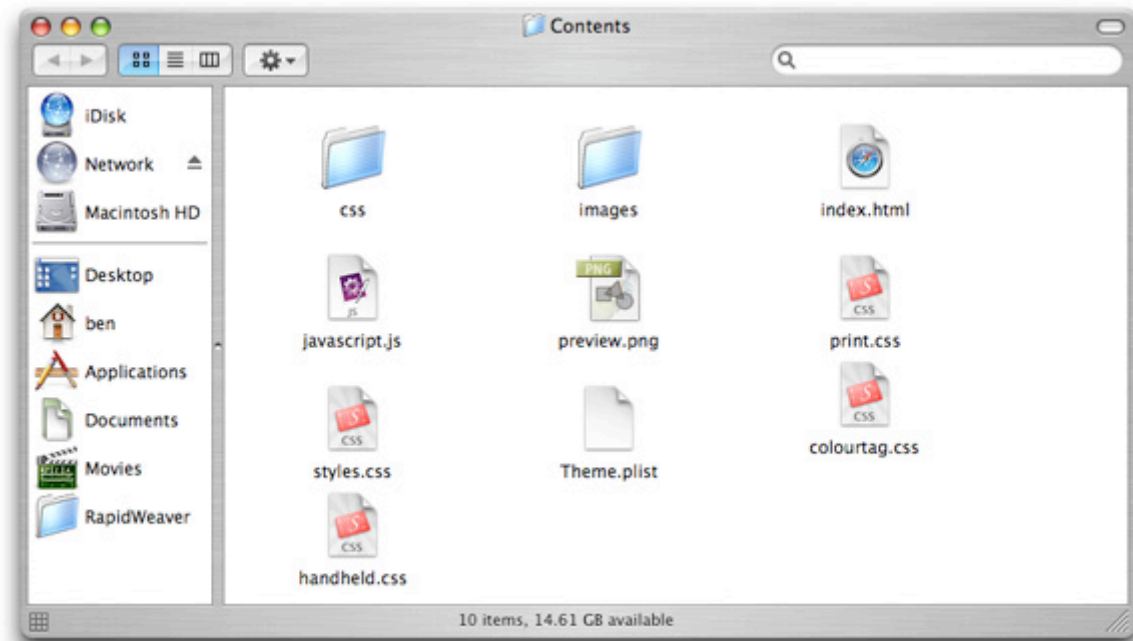
Once you have the theme installed, open the *Tutorial Site* project file and choose the tutorial theme from the theme drawer. Right click on the theme preview and select *Show Contents*.



This will reveal a *Contents* folder in the finder, inside are all the files required to construct a RapidWeaver theme. You should have the following files:

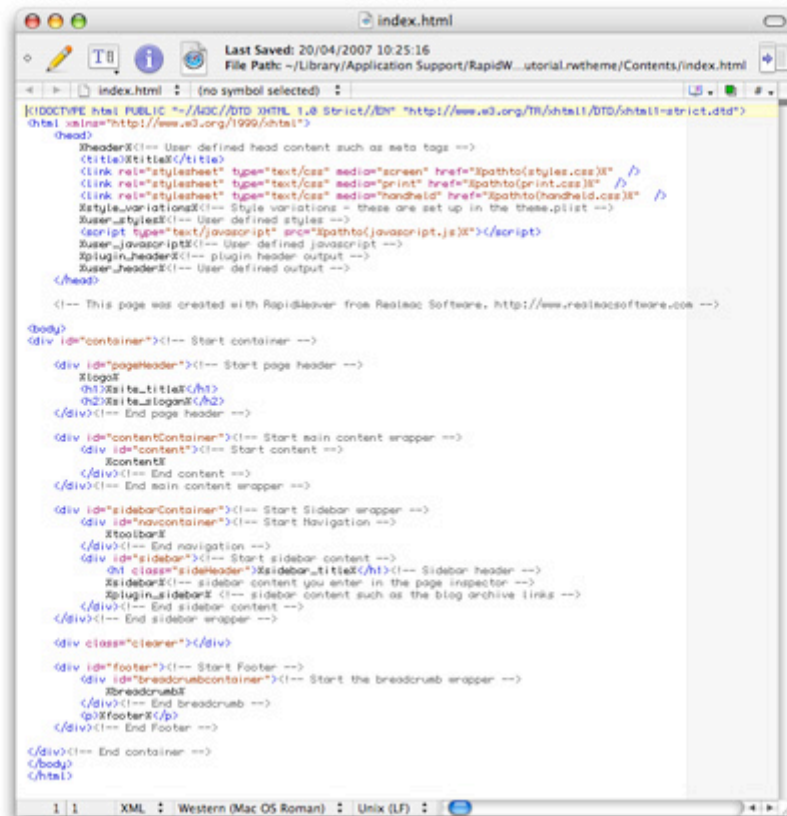
- index.html - template file
- styles.css - base CSS styling
- colourtag.css - base colour style variations CSS
- print.css - print CSS
- handheld.css - handheld device CSS
- javascript.js - required javascript
- images - where all images for the theme are placed

- css - theme variations CSS files are placed in here
- preview.png - the preview of your theme display in the theme drawer.



## The Template Page

The template page is the basis for all other pages to be generated from. Open *index.html* in your text editor. Open index.html TextWrangler (or similar) and you'll see that it is a standard XHTML page but also includes numerous RapidWeaver syntax tags.



```
<!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">
  <head>
    <!-- User defined head content such as meta tags -->
    <title><titleX</title>
    <link rel="stylesheet" type="text/css" media="screen" href="<pathto(styles.css)>" />
    <link rel="stylesheet" type="text/css" media="print" href="<pathto(print.css)>" />
    <link rel="stylesheet" type="text/css" media="handheld" href="<pathto(handheld.css)>" />
    <styleVariationsX<!-- Style variations - these are set up in the theme.plist -->
    <user_stylesX<!-- User defined styles -->
    <script type="text/javascript" src="<pathto(javascript.js)>"</script>
    <user_javascriptX<!-- User defined javascript -->
    <plugin_headerX<!-- plugin header output -->
    <user_headerX<!-- User defined output -->
  </head>
  <!-- This page was created with RapidWeaver from Realac Software, http://www.realacsoftware.com -->
  <body>
    <div id="container"><!-- Start container -->
      <div id="pageHeader"><!-- Start page header -->
        <logoX
        <h1><site_titleX</h1>
        <h2><site_sloganX</h2>
      </div><!-- End page header -->
      <div id="contentContainer"><!-- Start main content wrapper -->
        <div id="content"><!-- Start content -->
          <contentX
        </div><!-- End content -->
      </div><!-- End main content wrapper -->
      <div id="sidebarContainer"><!-- Start Sidebar wrapper -->
        <div id="navContainer"><!-- Start Navigation -->
          <toolbarX
        </div><!-- End navigation -->
        <div id="sidebar"><!-- Start sidebar content -->
          <div class="sidebarHeader"><sidebar_titleX</div><!-- Sidebar header -->
          <sidebarX<!-- sidebar content you enter in the page inspector -->
          <plugin_sidebarX<!-- sidebar content such as the blog archive links -->
        </div><!-- End sidebar content -->
      </div><!-- End sidebar wrapper -->
      <div class="clearer"></div>
      <div id="Footer"><!-- Start Footer -->
        <div id="breadcrumbContainer"><!-- Start the breadcrumb wrapper -->
          <breadcrumbX
        </div><!-- End breadcrumb -->
        <pageFooterX</p>
      </div><!-- End Footer -->
    </div><!-- End container -->
  </body>
</html>
```

### RapidWeaver Syntax Tags

The Syntax Tags found in the template file will be replaced by RapidWeaver with actual content, the tags can be placed anywhere in the template. A RapidWeaver syntax tag looks like this:

```
%content%
```

This is the tag for the main content of the page. A full list of tags can be found in the *Theme Syntax* document included in the *Theme SDK*.

### The pathto(...) Tag

All RapidWeaver syntax tags, with one exception, look exactly the same. The exception is the `%pathto(file.extension)%` tag. This tag is used for site consolidation. The `pathto` tag tells RapidWeaver that the file is located in the common site directory (`rw_common`) and therefore it places the folder path the file will be located in before the file name. For example, the `pathto` tag that links to the `styles.css` file look like this:

```
%pathto(styles.css)%
```

the published code will look something like this:

```
../rw_common/themes/realmacsoftware/styles.css
```

### ***CSS IDs and Classes***

All the syntax tags that are placed in the body of the template are wrapped in XHTML code referencing CSS classes. These XHTML “wrappers” allow us to style the content easily with CSS.

For example, the `%content%` tag is wrapped inside two XHTML tags:

```
<div id="contentContainer"><!-- Start main content wrapper -->
  <div id="content"><!-- Start content -->
    %content%
  </div><!-- End content -->
</div><!-- End main content wrapper -->
```

The `%content%` tag is directly wrapped inside the `content` CSS id class, which is in turn wrapped in the `contentContainer` CSS id class. It is wrapped inside two tags so that we can set an overall width and apply padding, this is needed due to inconsistent support of CSS across web browsers.

### ***IDs and classes, what's the difference?***

It's important to understand that there are two types of CSS elements that can be applied. One is an ID applied in XHTML like this:

```
<div id="my_id_name"></div>
```

An ID is styled in CSS like this:

```
#my_id_name {...}
```

The other is a class applied in XHTML like this:

```
<div class="my_class_name"></div>
```

A class is styled in CSS like this:

```
.my_class_name {...}
```

The important detail to remember is that an ID can only be used once on each XHTML page and is styled in CSS with an "#". A class can be referenced limitlessly and is styled with a "." (period).

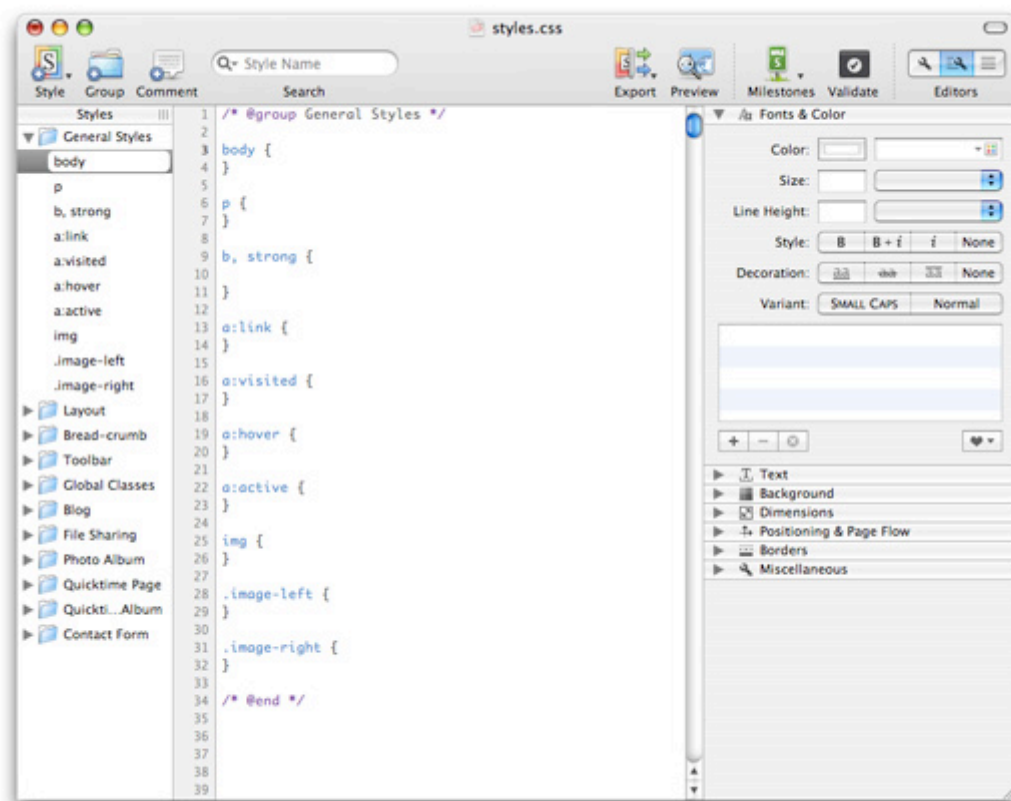
Each wrapper name used in the template page should be self explanatory but familiarise yourself with the `index.html` code as it will help you to understand the CSS section of this tutorial.

## Styling Your Theme

I will be using CSSEdit ([www.macrabbit.com](http://www.macrabbit.com)) to edit the CSS files, I would recommend you purchase a copy, however you can continue to use TextWrangler for this section.

This section covers CSS (Cascading Style Sheet) and how you can use it to style the overall layout. A full list of RapidWeaver's style classes with explanations can be found in the *CSS Styles* document included in the *Theme SDK* download.

### *Open the styles.css file*



### *Common Styling*

To add styling that will apply to your entire theme you can edit the *body* tag. This will style everything viewable in the browser window. This is where you should add common styles such as a base font size, font family, font colour and the line height. To do this add the following code:

```
body {  
  color: #333;  
  font-size: 0.85em;  
  line-height: 1.4;  
  font-family: Arial, Helvetica, Geneva, sans-serif;
```

```
}
```

This code may look complex at first glance, however it's actually quite simple. Firstly we tell the CSS we want to style the body XHTML tag so we simply write *body* followed by open and close curly brackets (`{... }`). Between the curly brackets is where the styles for each tag are placed.

To add a style property to a class you firstly write the style name, such as *color*, then a colon, then the style value (`#333` in this case) and then a semi-colon; `color: #333;`

Next we add a font size of *0.85em*. An em or 1em displays the font at the base size for that family. Each font family can vary so results from using em's will not always be the same. The advantage of using em's is that it is a relative size, not fixed like a pixel. This allows the text to be resized for each visitors requirements. For example, a sight impaired visitor will be able to increase the size of the font without any problems.

A line height of 1.4 is then added. This allows us to increase the height between each line of text giving our text a better flow and making it easier for the visitor to read.

Finally we add a font family. I've added a web safe family; *Arial, Helvetica, Geneva, sans-serif*. This way I can be sure my text will display the same, or a very similar, font on each browser. I have set three alternative fonts and a default sans-serif type face to use if the user has none of the previously specified fonts available.

### ***Top Tip***

**Instant Updates:** RapidWeaver's instant updates is a great feature because it keeps track of all the theme files you're using and notices when they get updated. When any file is updated RapidWeaver's preview will update; great for theme testing. Just leave RapidWeaver in preview mode and when you save your CSS you'll see RapidWeaver update instantly in the background.

### ***Link Styling***

Link styles are special because they have four different states; *link*, *visited*, *hover* and *active*. Each style is applied to the link based on the users action on that link. They are all quite self explanatory, *link* being the original state, *visited* being the visited state, *hover* being when the user place their cursor over the link and *active* being when the user clicks on the link.

If no styling is applied there is a set default styles that are applied. We'll want to personalise our links and to do that we'll add the following code:

```
a:link {
    color: #82aa15;
}
a:visited {
    color: #55700f;
}
a:hover {
    color: #a2db36;
}
a:active {
```

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

This sets a different colour for the *link*, *visited* and *hover* states and sets the text decoration to none for the *active* state. By default the link styles have an underline as their decoration. Try rolling over and clicking one of the links in RapidWeaver, you'll notice the styles change.

### **Site Width**

Add the following code to the #container tag:

```
#container {
    width: 720px;
    margin: 0 auto 0 auto;
}
```

The #container tag is set in the template page, it is wrapper for the entire site which is used for setting the overall width (*720 pixels*) and position of the site.

Save the "styles.css" file and you'll see RapidWeaver update in the background. If you now stretch the RapidWeaver window across your screen you will see that the page content is centred on the page. This is achieved by the code you just added. The width property is obvious, however the margin property is not.

The margin property we have added tells the browser we want the site to be centred.

In CSS you always specify properties such as margins and padding in the same order; *top right bottom left*. So our code for the margin set the top and bottom to 0 margin and the right and left to *auto*. The auto margin tells the browser to set half the browser window with, after it has taken the container width in to consideration, to the left and right margins therefor making the content centred.

We can condense the code down a little. If the top and bottom margins are to be the same and the left and right are to be the same then the following code can be used

```
#container {
    width: 720px;
    margin: 0 auto;
}
```

This way the margin properties for bottom and left are inherited. If we were to simply put *margin: 0*; it would apply to all four sides. The same applies for padding.

### **Content and Sidebar**

Our tutorial site currently displays the sidebar content below the main content. For this tutorial we want the sidebar to be on the right side of the content.

To get the sidebar to display along side the main content, first we need to set a width to the content container and tell it to be on left side:

```
#contentContainer {
    width: 500px;
    float: left;
```

```
}
```

Next we need to set a width to the sidebar and tell it to display on the right side:

```
#sidebarContainer {  
    width: 220px;  
    float: right;  
}
```

To get the content and sidebar to display next to each other the *float* property is used. I won't go into detail about the float value but I suggested reading this W3C document: <http://www.w3.org/TR/CSS21/visuren.html#propdef-float>

Save your CSS and notice that the content and sidebar are now displayed side by side.

It's important to remember that the combined width of the *contentContainer* and *sidebarContainer* should be equal to or less than the width of our *container*. If the combined width is more than the container, the sidebar would be displayed below the content.

The breadcrumb and footer should be displayed below the content in the theme, however at the moment they are being displayed in the sidebar area. To fix this problem we need to make sure they clear both the content and sidebar. We do this with a clearer class set-up in the template file. Add the following to the clearer class:

```
.clearer {  
    clear: both;  
}
```

### ***Content Padding***

Adding some padding to our content area is a simple task, simply add the following to the *#content* like so:

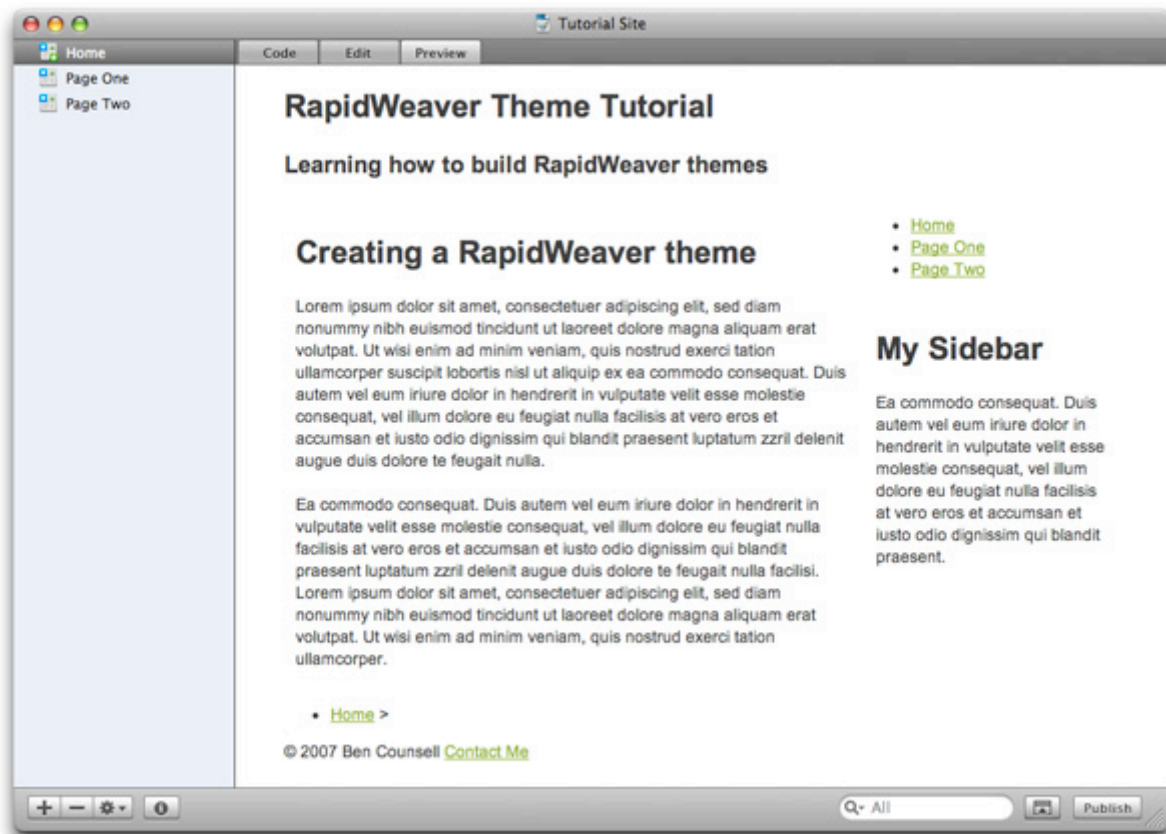
```
#contentContainer #content {  
    padding: 10px;  
}
```

This is slightly different to other classes we have styled before. This time there are two tags on the same line; *#contentContainer #content*. This tells the browser that the *#content* tag is wrapped inside the *#contentContainer*. This is important to remember as it allows you to style, for example, a link differently in the main content area than in the sidebar.

Add the same padding to the sidebar wrapper:

```
#sidebarContainer #sidebar {  
    padding: 10px;  
}
```

Save your CSS and you'll see your site is starting to take shape nicely!



## Page Header

The page header is the area of the site where the title, slogan and logo is placed so it's usually important to make it stand out from the rest of the site. To do this first add some padding:

```
#pageHeader {  
    padding: 10px;  
}
```

That's nice and simple, next lets add a background colour:

```
#pageHeader {  
    padding: 10px;  
    background-color: #82aa15;  
}
```

## Title and slogan

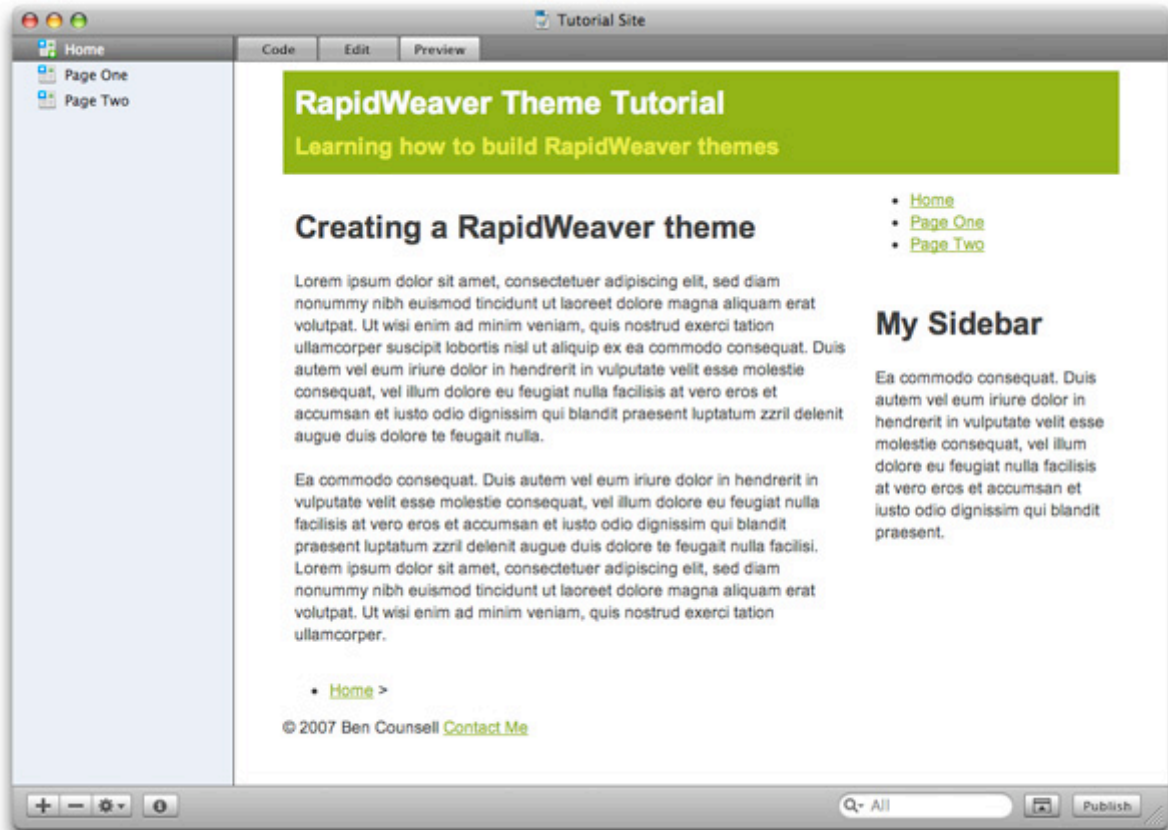
Next change the colour and margins of the site title and slogan like so:

```
#pageHeader h1 {  
    margin: 0 0 5px;  
    color: #fff;  
}  
  
#pageHeader h2 {
```

```
margin: 0;
color: #e4ee3b;
}
```

The site title and slogan tags are inside a *h1* and *h2* tag respectively in the template page, these tags were nested inside the *pageHeader* id. Therefore we are telling the CSS to only style the *h1* and *h2* tags this way for the *pageHeader*. The *h1* and *h2* tags for other areas of the site remain untouched.

Save your CSS and your theme should look like the preview below.



## Navigation Styling

Navigation styling is an important part of your theme design, it allows users to easily navigate through your site.

Firstly we'll add some padding to the navigation wrapper to match the content and sidebar:

```
#navcontainer {  
    padding: 10px;  
}
```

The `%toolbar%` syntax tag outputs the menu for the site. RapidWeaver outputs this as an unordered list. The unordered list (`<ul>`) is a standard XHTML tag which contains list items (`<li>`). The output for the site menu looks like this:

```
<ul>  
    <li><a href="index.html" rel="self" id="current">Home</a></li>  
</ul>
```

New navigation links will be added as list items (`<li>...</li>`), fortunately RapidWeaver creates all of our navigation code for us, all we need to worry about is the styling. The other important piece of code to take note of is the `id="current"` which is applied to the link tag. This style is only applied to the current page and allows us to style the current page link differently from the rest of the site. This is important to let visitors to the site know which page they are currently on. We'll look at styling current links in a few minutes.

### Default Styles

As mentioned before, standard XHTML tags have default styling values applied to them. You may have noticed this already because the *Home* navigation item has a circle to the left of it. This is a default value of unordered lists, to modify this and some the default margins and padding add the following code:

```
#navcontainer ul {  
    margin: 0;  
    padding: 0;  
    list-style-type: none;  
}
```

To style the navigation links add the following code:

```
#navcontainer a {  
    display: block;  
    padding: 5px;  
    background-color: #82aa15;  
    color: #fff;  
    text-decoration: none;  
    margin-bottom: 2px;  
}
```

There are many CSS properties here that we have used before and you should be getting familiar with these by now. The `display: block;` is a new property, it allows you to set the display value of the item. For example you can set `display: hidden;` and it will hide the element. In our theme though we want our links to display as a *block* taking up the full width

of the navigation wrapper.

We'll use the *hover* state of the link to change some link styles when the users' mouse hovers over the link:

```
#navcontainer a:hover {  
    background-color: #e4ee3b;  
    color: #82aa15;  
}
```

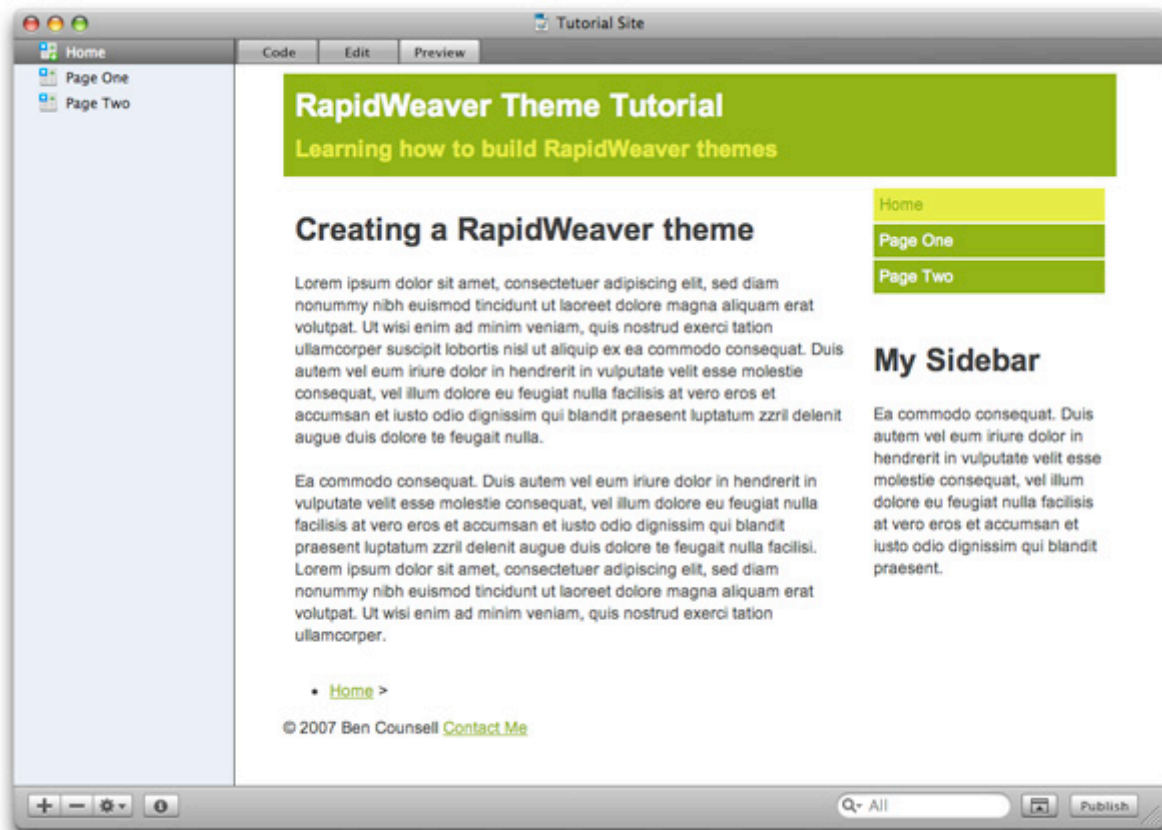
Save your CSS and view your site in RapidWeaver. Try hovering over the navigation links.

### ***Current Link Styling***

As mentioned above, we have an id named *current* that we can use to style the current page link. To do this add the following:

```
#navcontainer #current {  
    background-color: #e4ee3b;  
    color: #82aa15;  
}
```

Your theme should now look like the preview below.



## Footer Styling

For the final part of this section we will look at how to style the footer. To separate the footer area from the content we'll make the text smaller, centered and add a border, this is achieved like so

```
#footer {
    font-size: 0.9em;
    text-align: center;
    padding: 10px;
    border-top: 2px solid #82aa15;
}
```

The breadcrumb links still require a little styling. The breadcrumb is a trail of links through the site to the page being viewed. For Example if you were filling out a support contact form your breadcrumb might look like this:

**Home > Support > Contact**

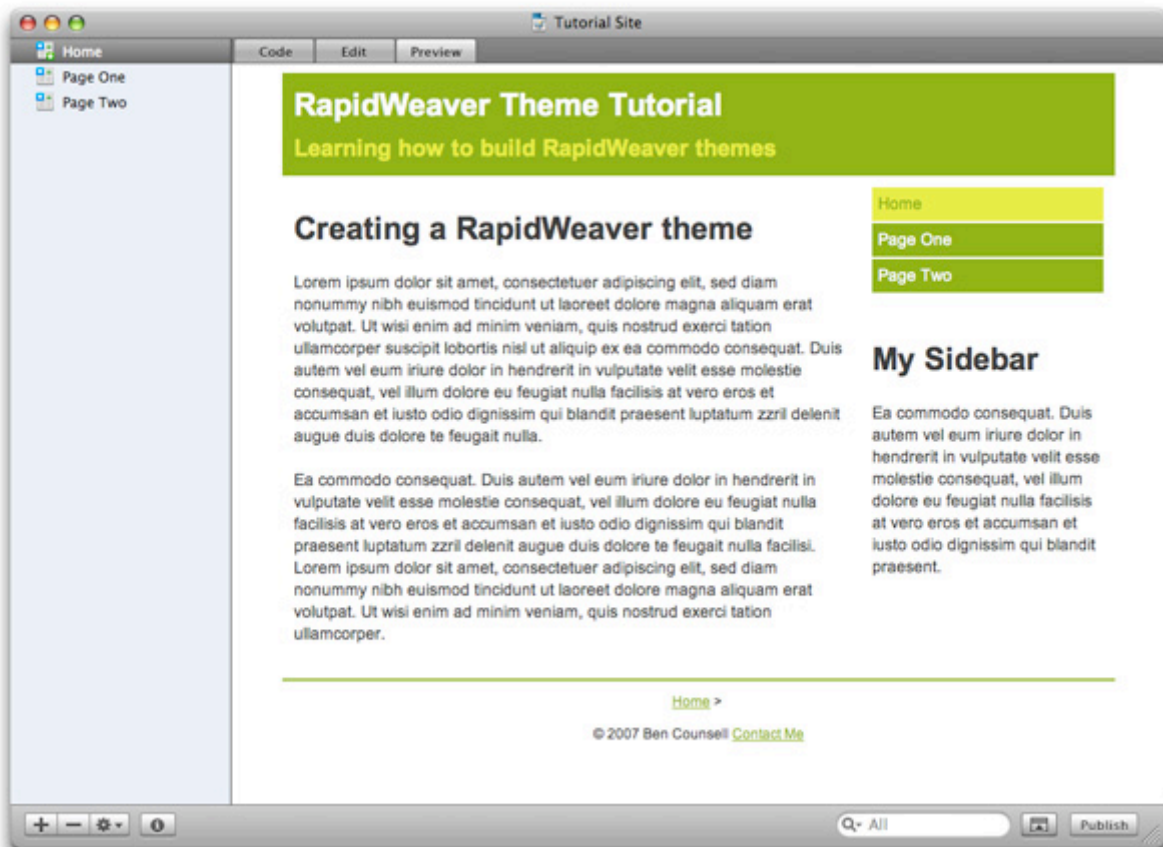
In the same way with the navigation links, the breadcrumb is output as an unordered list. To remove the standard styling from the unordered lists add the following to your CSS

```
#breadcrumbcontainer ul {
    margin: 0;
    padding: 0;
    list-style-type: none;
}
```

Finally make the breadcrumb items display inline:

```
#breadcrumbcontainer li {
    display: inline;
}
```

Save the *styles.css* document.



## Conclusion

This tutorial is aimed at giving you a basic understanding of how RapidWeaver themes are constructed. This tutorial only scratches the surface of what can be achieved, in future tutorials we'll look at styling RapidWeaver's page types and adding more advanced features to your themes.

For further reference of RapidWeaver's syntax and CSS styles read the *Theme Syntax*, *Theme.plist Syntax* and *CSS Styles* documents included in the Theme SDK download.

If you're feeling a little overwhelmed by this tutorial I would suggest purchasing a good web standards and CSS book. You can also visit our forums at <http://www.realmacsoftware.com/support> and read through the many user tutorials and of course ask questions.

### *Useful CSS Links*

CSS properties: <http://www.w3.org/TR/CSS21/propidx.html>

Web safe fonts: <http://www.ampsoft.net/webdesign-1/WindowsMacFonts.html>