

net tuts+

presents



FROM

PHOTOSHOP

TO HTML

How to slice your
designs like a pro

Jeffrey Way

ROCKABLE*

Rockablepress.com
Envato.com

© Rockable Press 2010

All rights reserved. No part of this publication may be reproduced or redistributed in any form without the prior written permission of the publishers.

3

Foreword	5
An Introduction	6
<i>What This Book Includes</i>	7
<i>What This Book Assumes</i>	7
<i>What This Book Does Not Assume</i>	8
<i>Tools of the Trade</i>	8
Coding The HTML	12
<i>Coding The HTML</i>	13
<i>Creating the Basic Shell</i>	15
<i>The Markup</i>	16
<i>Markup Complete!</i>	27
Slicing the PSD	32
<i>The Plan</i>	33
<i>Photoshop</i>	34
<i>The “Save for Web” Pattern</i>	36
<i>Best Practices for Image Types</i>	39
<i>CSS Sprites</i>	41
<i>Revising our HTML</i>	46
Coding the CSS	49
<i>CSS Resets</i>	50
<i>What’s the Solution?</i>	51
<i>DEFAULT.CSS</i>	54
<i>General Styles</i>	56
<i>Container</i>	58
<i>Header</i>	58
<i>Styling The Main Banner</i>	62
<i>Fahrner’s Image Replacement Technique</i>	64
<i>What’s The Difference Between Absolute And Relative Positioning?</i>	65
<i>Main Content</i>	68
<i>Working Example: Columns</i>	69
<i>Here’s The Thing about Floats</i>	75
<i>The Three Headings</i>	78

4

<i>Blockquote</i>	81
<i>CSS Shapes</i>	83
<i>Recent Portfolio</i>	85
<i>Before</i>	86
<i>After</i>	90
<i>About Page Header (subpage)</i>	101
<i>Home Page Header</i>	101
<i>The Body ID Trick</i>	102
<i>Work Page</i>	104
<i>Contact Page</i>	111
<i>Cufón Font Replacement</i>	118

Compensating for Older Browsers **123**

<i>Cringe Time</i>	124
<i>Tools</i>	124
<i>Why is “Having Layout” Necessary?</i>	126
<i>Targeting Specific Versions of Internet Explorer</i>	127

Bonus Chapter: A Splash of jQuery **134**

<i>The Script</i>	135
<i>Decoding the Script</i>	137

Conclusion **140**

Appendix – Further Study **142**

About The Author **143**

Foreword

I'm sure you've heard it before. "CSS is easy. Anyone can do it!" Granted, it may not be rocket science, but CSS is powerful enough to make you rip your hair out.

On a daily basis – if you listen closely – screams can be heard around the world from coders unsuccessfully attempting to force a design into position. It's not just a matter of learning the language; memorization is an easy task. The scream inducing points occur when you view your site in ten different browsers, all of which render your site at different levels of consistency. When such things occur, how do you fix them?

This is what separates the amateurs from the pros. By the end of this book, you'll be a member of the latter.

It won't be an easy task. As a matter of fact, I'm going to teach you to use techniques that seemingly make zero sense. Aren't you glad you purchased this book? All kidding aside, you're going to find that, just because it should work, doesn't mean it will. Just because your design looks great in Firefox, doesn't mean it will in Internet Explorer – in fact this is typically the case. To compensate, we'll be researching the unique "deficiencies" in each browser that will undoubtedly prove to give you trouble. When you finish the last page, you'll have the tools to battle the forces of evil and wrangle browsers – even those that are a decade old – to work for you, not against.

Jeffrey Way

AN INTRODUCTION

What This Book Includes

Packaged with this book, you will find:

- 1) A **directory of example files** made available for your reference. These files correspond to the example designs and HTML used in this book, and consists of: **Photoshop files for the Light Box Design and HTML / CSS files for the Light Box Theme**. These files and themes may be used freely in your projects, both commercial and non-commercial. However, they may not be redistributed or resold in any way. As you work through this book you may choose to either construct your own set of files from scratch, or use the example files as a guide.
- 2) A series of **six screencasts** covering the whole project from beginning to end.

To access these files, please use the links provided on the last page of this book.

What This Book Assumes

You should have at least an elementary understanding of HTML and CSS. You should be familiar with HTML tags and the most common CSS properties. For example, it will be assumed that you understand the syntax, and the relationship between the following HTML and CSS snippets:

HTML

```
<div id="myDiv">  
...some text goes here  
</div>
```

CSS

```
#myDiv {  
  width: 500px;  
  margin: auto;  
  background: red;  
}
```

Additionally, you should have a basic understanding of Photoshop.

What This Book Does Not Assume

Nothing more will be assumed beyond a basic understanding of the syntax for these two languages.

Tools of the Trade

When converting a PSD to HTML and CSS, you'll be most efficient if you have the right tools at your disposal. Luckily, other than Photoshop, there are a plethora of programs and extensions available to you, most of them free. In this book, we'll be using the following:

Komodo Edit

https://www.activestate.com/komodo_edit/downloads/

There are a variety of free code editors for both the Mac and PC. A simple Google search will return several options. One that will beautifully serve users of both platforms is Komodo Edit.

Web Developer Extension for Firefox

<https://addons.mozilla.org/en-US/firefox/addon/60>

This nifty extension will allow you to edit your CSS inside of the browser, disable JS, apply rulers, validate the HTML and CSS, to name a few. It's an essential tool that nearly all developers have installed. You should as well.

IE Tester

<http://www.my-debugbar.com/wiki/IETester/HomePage>

Unfortunately, even after a decade, we must still compensate for older browsers – namely Internet Explorer 6 – when converting designs. Most likely, you'll have IE7 or IE8 installed though. In order to preview your designs in this browser, one option – if on the PC – is IE Tester. It allows you to view any site in IE5 – IE8.

BrowserShots.org

<http://browsershots.org/>

If you work on a Mac (or need to test in a larger variety of browsers), a second option is to use a site called Browsershots, to take a snapshot of your design in all of the browsers.

VMWare Fusion

<http://www.vmware.com/products/fusion/>

Especially in the design world, Mac users are quickly becoming the majority. Though Browsershots is definitely helpful, sometimes

the best option, when needing to test in IE6, is to use a program like VMWare Fusion. As they put it, “Windows is even better on a Mac!”. Once loaded, you can then easily run IE Tester.



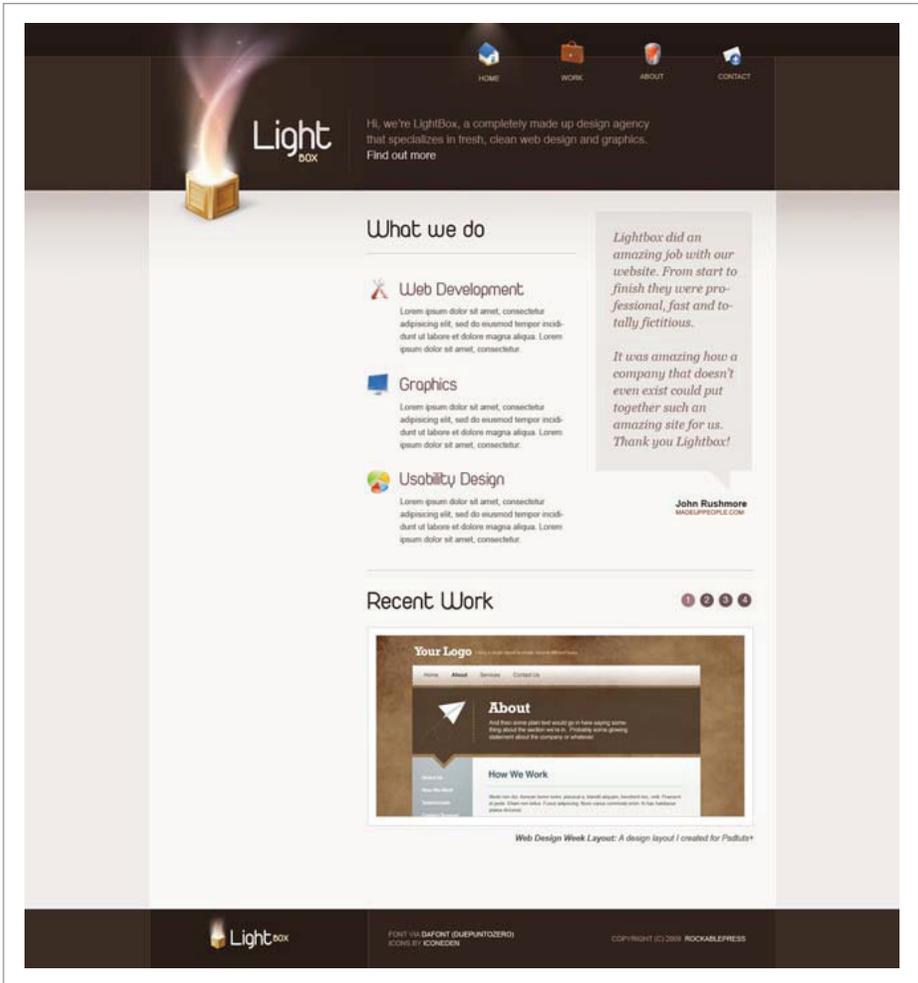
Coding The HTML

While it might seem natural to immediately begin working on the visuals of our website, this actually couldn't be further from the truth. Instead, we must first build our base, or the mark-up, and only once it's been completed can we move on to the visuals.

In this section, we'll analyze the design, and then translate the structure to HTML mark-up.

Coding The HTML

Consider the following layout:



We will be converting this exact design to a complete HTML / CSS website! When viewing such a design, we must begin to mentally format our layout from an HTML point of view. After a bit of observation, you might imagine the following layout:



You should begin thinking this way when viewing every design. This layout is essentially divided into three sections:

1. Header
2. Main Content
3. Footer

You'll find that most sites will have a similar structure. Some might have a few more columns, but in general, you'll always find these three components.

Creating the Basic Shell

Now that we've visualized our design from a coder's point of view, let's create the basic HTML formatting.

We begin by creating a "container" division. That way, if necessary, we can position our entire website simply by altering one div.

```
<div id="container">
  ...stuff
</div>
```

ROCK* TIP

Think of a div as a box. Within this div, we can insert a bunch of stuff (elements). Then, if we need to move all of that stuff around, we simply pick up our box and reposition it!

Next, add a few more divisions. Ignoring the specifics and focusing solely on the wrapping elements, let's add our three components:

```
<div id="container">
  <div id="header">
    header content goes here
  </div>

  <div id="main">
    main body goes here
  </div>

  <div id="footer">
    footer info goes here
  </div>
</div>
```

Referring back to our design, we could then begin inserting the correct elements: an image here, a blockquote there. If you've never converted a PSD before, you're already on your way! **It's as simple as visualizing the layout from a coder's perspective, and creating the markup to match it.**

The Markup

Let's now take the process a step further. Within our `#main div`, we'll need to insert additional wrapping divs for the "What We Do" section as well as the sidebar.

If we wanted to, we could simply name the two divs, "whatWeDo" and "right-sidebar." However, this method could potentially cause problems. What if, down the road, we instead decide to display a list of recent products? In that case, we'd have to rename this div, and all of its references within the stylesheet. Obviously, "whatWeDo" wouldn't make any sense!

What if we decide to switch the sidebar to the left side? As a general guideline, remember to name your elements according to what they are, and avoid presentational specifics. This is referred to as "semantics." For example, consider the `<i>` and `` HTML tags. They essentially perform the exact same task: making text italic. With that said, their meanings are quite different. The `<i>` stands for "italic," while `` stands for "emphasis." Notice the difference? One element refers to the styling (or presentation), while the other simply describes what text should be – emphasized.

ROCK* TIP

As a rule of thumb, if you have to edit both the HTML AND the CSS when adjusting the styling of your page, you're doing it wrong.

Always separate presentation from content.



Returning to our project, we'll simply use "primaryContent" and "secondaryContent."

```
<div id="container">
  <div id="header"></div>

  <div id="main">
    <div id="primaryContent"></div>
    <div id="secondaryContent"></div>
  </div><!-- end main-->
</div><!--end container-->

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



Also, take note of how our footer div is a sibling of our container div, not a child (an element that is nested inside of another). The reason is because we want the brown background to extend for the entire width of the user's browser window. This cannot be accomplished when nested inside a div with a stated width. Don't worry if this is confusing; we'll discuss it more in later sections.

ROCK* TIP

Do not touch your CSS file until the markup has been completed.



Header

Now that we have our basic structure, let's begin fleshing out each section. We'll start with the header.