

frugal **THEME HANDBOOK**

Table of Contents:

Introduction:

Chapter 1: The Basics of Web Design

Chapter 2: Wordpress Themes

Chapter 3: The frugal Wordpress Theme

Chapter 4: Paint By Numbers | It's Time To Customize!

Chapter 5: Custom Logo, Avatar, Favicon and Positioning

Introduction:

If you're reading this then I'm assuming two things. One, you just purchased my frugal theme, and two, you're passionate about blogging. As far as the second assumption, I can totally relate! I absolutely LOVE all

things blogging! From writing, to promoting, to design, to social media and my favorite part...community. But to enjoy these wonderful benefits of blogging we much learn to never give up. The fact is, successful bloggers aren't just great writers and promoters, but they have a certain amount of perseverance that keeps them pushing ahead, even when things seem futile. When the majority of used-to-be-bloggers have already dropped out of the race, this special breed of blogger is still trucking along, keeping a strong, steady pace. There's only one way to become a successful blogger and that is to keep moving forward (oh, and to write some seriously solid posts! :-)) But enough about that...

The main focus of this handbook is to make sure that you have the right tools and know-how to be able to mold your frugal theme into what it needs to be to best suit your blogging needs. You may have heard me say this before, but customizing your frugal theme is kind of like paint by numbers. By adding specific notes in the code and then creating a kind of map in this handbook, I've made the customization process about as simple and straight forward as I possibly could. I want the focus to be on WHAT you want your theme to look like, not HOW you're going to do it.

So I hope you enjoy reading through this handbook and learn a lot. I've tried to make it both easy to understand and light on the brain. The last

thing I want to do is bore you to tears, so I hope I've managed to keep the technical jargon to a minimum. And when I DO use technical terms, you can assume that I felt it necessary to do so. Otherwise, it should be a pretty lite read.

Chapter 1: The Basics of Web Design

As you probably already know, a blog theme is just another website template. Pages of code that sit on web servers and tell browsers what to put on their pages, these themes are nothing but plain old websites. So learning about web design is leaning about blog theme design. So I wanted to start by sharing so general tips about what a solid site should do to make the most of its online real estate.

Framework:

First and foremost, a website design is just a kind of framework for the content it presents in its boundaries. From the header where the logo/title sits, to the navigation bar that is attached in some way, shape or form to any sidebars, feature sections and the final footer to close the circuit. All of this, surrounding the main content area where you're focused voice is present. This framework is the glue that holds your website together.

Starting with the header: An effective header is one that is just the right size to make visible your name-brand, but not too much to burst your readers personal bubble. It should fallow the theme of the rest of the site, but carry just enough difference that it stands out and draws those new eyes into itself. You want a new visitor to be drawn to it, but not so much that they can't look away.

The navigation bar: I personally believe that this is the most important part of the framework. Acting as a kind of anchor, the navigation bar (or navbar for short) becomes the boundary between your header and the rest of your content (this assumes it's on top and not on the side, but that's an assumption I'm going to make in this handbook). It holds the keys to each page on your website and allows instant access, no matter where you go. The header is important, but without the navbar, it would feel naked and out of place.

The key to an effective navbar is simple. Use JUST the necessary amount of pages to it and it will do its job. Too many and they become lost in themselves, doing nothing more than confusing your short attention spanned reader.

Sidebar: The sidebar is your sub-content area. Kind of a mix between the navbar and your main content area, the sidebar both carries access to other locations as well as house content you want readily available with just the slightest glance.

A solid sidebar is one that is JUST big enough to house your necessary links, buttons, forms and images. Keep it simple, but not bland. You want to draw your readers eye, but not distract from your main content.

Speaking of your main content area: This should be the largest piece of land on your website, housing the focus and point of your pages. White or light background is ideal, with dark colored text. Clean is always good and the rest is up to your own grammatical skills.

The footer: The final piece to the puzzle, the footer closes the loop to keep the circuit circling. There are two kinds of effective footers and both are about as opposite as they come...

One is the simple cut off point. This should be almost invisible, but just enough to house the credits and bounce your readers eye back up to the top. The second kind is one that houses much more than the credits. Used as a feature section, this kind of footer is almost a kind of bottom sidebar. A

great place to put some 'bonus' links for your readers to feast on, this can be a solid way to extent your website just a bit further. Just be careful not to over do it. Too much clutter in the basement of your site and you'll lose your visitors in the mess of pictures, links and who knows what else.

So keep it clean either way and try and stick to the JUST ENOUGH rule.

Feel:

The feel of your website should be just what it needs to be to draw your reader in. If it's a news site, it should feel informational. If it's about art, it should feel creative. Whatever your content is trying to convey, your website should back this up with the right layout.

But don't mix this up with the idea that the framework should be decked out. The fact is, what you put INSIDE your layout should determine the look and feel, not so much everything else. I mean sure, it doesn't hurt to have a unique, attractive custom logo and the colors DO make a difference, but the words and images that fill the insides, those are what will solidify your message. The framework should be just that...a FRAME. Just like too much picture frame can take away from the picture, so too can your over-the-top web design distract from the content inside.

Keeping people coming back:

The fact is, a new visitor isn't your bread and butter. It's only when that person re-appears at a later time that you know you've done something right. No amount of theme customization will be enough to ensure the masses will come back, time and time again. It's up to YOU to fill the walls of your website with words and images that inspire a return. Successful online activity may start with a solid frame, but this quickly transitions to its rightful place, in the hands of your voice.

Chapter 2: Wordpress Themes

Now it's time to get a bit more specific and focus on Wordpress. Let's face it, we've got it easy! Wordpress does so much of the work when it comes to dealing with the many different web duties, that we can just focus on the simple layout and design of our theme. This allows more of the non-tech person to be able to dip their toes into the sea of simple web design. No longer do we have to pay thousands of dollars for a unique, custom look. These days, with the power of Wordpress, the opportunities are endless.

OK, enough butt kissing, let's get to the themes themselves. Wordpress

themes are made up primarily of two kinds of files...the PHP file and the CSS file. These two files types work together to create both the specified function and look of the theme.

First, the PHP file:

The files that provide functionality to the Header - Footer - Sidebar - Comments - Archives - etc... are all written in PHP. Just a name for a type of computer code, PHP, like any other type of code, is just some letters, numbers and words that form functions in a text file. Sounds complicated, but it's really not.

The CSS file:

With most Wordpress themes, there is only one CSS file which makes up the Stylesheet. This is just a text file that creates the look of your theme. It just tells all the PHP files how to look. So, for example, if you have a header.php file, you will most likely have a section in your style.css file called header. This section will tell the PHP file to have a certain color, type and sized background. I may say what kind, size and color the font will be that may be present in that header. So the PHP file may hold the code that says WHAT that particular part of your theme will do, but it's the CSS file

that tells that PHP file HOW to look while doing it.

Images:

The only other part of a basic Wordpress theme is a directory called images. This folder houses the images that may be present in the theme's layout. It's THAT simple.

Customizing a Wordpress theme:

I'll get into the details in the next chapters, but I just wanted to share some universal facts about customizing Wordpress themes. There's no magic, no tricks. It's just a matter of changing certain letters, numbers and/or words around to make it do different things. Often times it is as simple as changing a single digit. Refresh your web page and BINGO! you have a different sized something or a shade of a different color. So don't let this idea of Wordpress theme customization scare you one bit. It's not rocket science, just simple logic.

Chapter 3: The frugal Wordpress Theme

Now that I've talked about both the basics of web design and then the basics

of the average Wordpress theme, I want to now touch on what makes the frugal theme unique.

First, the name:

I decided to call it frugal for two reasons. For one, I already had a web design business called Frugal Site Design, so it seemed fitting. But the other, more important reason, is because I wanted to convey the focus of its special framework. I wanted a theme that was absolutely as clean and simple as possible, but with functionality and style in mind. Thus, frugal made the most sense, to represent the JUST ENOUGH mentality.

The Code:

This is where the simplicity comes in. I created the most stripped down, minimalistic Wordpress theme code that I possibly could, without losing the features and functionality that I deemed worth the text space. The point of this was to not only allow for easy customization on your part (less code clutter makes for an easier time finding and changing the desired text), but I also knew that speed and reliability are also products of the simplicity mindset.

The Design:

Also sticking to this frugal state of mind, I designed the layout to be as bare bones as possible, while still maintaining an attractive style.

One reason for this all out speed. Without displaying a single image in its stock form, the frugal theme is able to load much faster than many bloated themes that are out there today.

Next, I wanted to give you amazing writers out there a theme that is ALL ABOUT THE CONTENT! For those who want ALL the attention on their precisely chosen words, the frugal theme is absolutely PERFECT.

Finally, I wanted to create the perfect blank slate for you creative types, who want to build that perfect blog design. With the frugal theme, you are given just that. Clean code and a clean, minimalist design that is waiting for your digital paint brush to swipe its first stroke.

The Support:

Through this handbook and the frugalTheme.com blog, I've created a support system that focuses on taking you WAY past the first stage of downloading

this theme. I wanted to bring forth a blogging product that not only gives you the tools, but the know-how to utilize them. So between this book and the blog I hope to bring you the necessary ability to finally create the look, feel and functionality of your perfect blog theme!

Chapter 4: Paint By Numbers | It's Time To Customize!

First, let me start by saying that the frugal Wordpress theme is quite capable in its stock form. When I created it, I wanted to make sure that you would have something worthy of your awesome content, right out of the gate. And for some, this super clean and simple design is PERFECT for their blogging needs. So don't feel you have to change ANYTHING if your goal is simplicity. Just enjoy the theme and blow us away with your blogging. Or, if you just want to make a few simple tweaks, jump to the necessary sections and learn how to do it. But if you're all about changing things around, read on and learn how simple it is to completely customize your frugal Wordpress theme.

The Basics:

Color: One basic thing you'll need to learn is that the colors in these coded

files are represented by a # sign, followed by 6 digit letters and/or numbers. White, for example, is #FFFFFF and black is #000000. So if you see a color that you want to adjust (whether it be a text color or a background color or whatever), you will need to find the right code to represent it. But don't worry, this is super simple. There are different ways to find this out, but one easy way is to go to this website <http://html-color-codes.info>. There you will be able to pick your desired color and then copy the code that represents it, ready to paste it into the area of desired adjustment.

Editing the CSS file: Most, if not ALL of your tweaking will be done in the one and only CSS file. It's called style.css and it's located in the main directory of the frugal theme.

The text inside this file is mostly broken down into categories (like Header - Navbar - Content - Sidebar - etc...). Inside each of these categories is clumps of text that range from 2 or 3 lines, up to maybe 5 or 6. These clumps are then separated by a single space (depending on how you're viewing the file).

The information in each clump is focused on specifying the look of your

theme. A typical grouping of text in the style.css file will look like this:

```
body {  
    background: #FFFFFF;  
    width: 960px;  
    color: #333333;  
    font-size: 12px;  
    font-family: Arial, Tahoma, Verdana;  
    margin: 0px auto 0px;  
    padding: 0px;  
}
```

As you can see, it starts by specifying WHAT it's focusing on (in this case, the body of the theme). Next, it lists different visual attributes and then specifies their look. I'll just go through this one piece-by-piece to help you understand the way it works.

- **background:** This is the background color of whatever area we're styling. Since it specified 'body', then this would be the background of the themes body. As you can see, this shows that the body is currently #FFFFFF or White. If you wanted to change the color of the themes background, you would just change the FFFFFFFF part to whatever 6 digit color code you prefer. So if you wanted it to be a certain shade of blue, you might use this code #045FB4 (remember to put the # symbol in front of the 6 digit number).

- **width:** So now we are specifying the width of the body of the theme. In stock form, the frugal theme is 960px (or pixels) wide. To change this, just adjust the number accordingly. Just know that if you change this number, you may have to change others for it to look right (I'll get into this later, so don't worry about it right now).
- **color:** When a wordpress style.css file just says color, it usually refers to text color. So this is saying that the overall text color for the body of your theme is #333333 (a very dark gray). Once again, to change this, just change the number and leave the # symbol in front of the new set of numbers.
- **font-size:** Pretty straight forward, this is just saying that the font size for any text in the themes body (unless otherwise specified) is going to be 12px. Change 12 to 14 and you will have a bit larger text size. Change it to 10 and it will be smaller. Very simple.
- **font-family:** This just specifies the style of font in the body of the theme (once again, unless otherwise specified). So why are there 3 fonts specified? The fact is, not all web servers (the computers that host your website) have the ability to show every font specified. Because of this, it is common practice to give the server more than one possibility. It first tries to show the first font. If it can, then it does. If it can't, it tries the second and then the third (if available) if necessary. So if you only give it one option and it is unable to show

that one particular font, it just won't show anything at all. This, as you know, could be a BIG problem. The two main font groupings in the frugal theme are Georgia, Times New Roman; and Arial, Tahoma, Veranda; The first is a more classical Times type of style and the second, well, is Arial...simple and plain. So to change your font, just change the grouping to your desired look.

- **margin:** This specifies the general location of, in this case, the body. You will find this in 3 forms: 0px; or 0px auto 0px or 0px 0px 0px 0px; They would all work and accomplish similar goals, but here's the basic differences. 0px, if changed, would change the margin for top, bottom, left and right. So if you made the number 20, then that would give the object a margin of 20px on all 4 sides. So 20px; is the same as saying 20px 20px 20px 20px; So if you wanted all the sides the same, it's simpler and cleaner to say 20px; but if you want to give any one or more sides a different margin size, you would need to use 20px 20px 20px 20px;. So what about the 'auto' one? This is rarely used in your frugal theme, but is done so for the body because the 'auto' part is what makes the body of the them automatically center in your web browser. Otherwise you would just use the other 2 variations.
- **padding:** Similar to the margin, it can use either of the 3 variations of specificity (0px; or 0px auto 0px or 0px 0px 0px 0px;) to move the

object around. But the padding is what specifies the position INSIDE the margin. So if you were to say 20px; you would be telling the object to sit exactly 20px from all 4 sides of the specified margin (if this is possible). So it's kind of specifying the buffer space.

- **margin AND padding:** If you are faced with the 0px 0px 0px 0px; all you need to know is that the numbers start at the top and move in a clockwise direction. So the first 0px is specifying the number for the TOP of the object. The second 0px will adjust the spacing for the RIGHT SIDE of the object. The third 0px focuses on the BOTTOM of the object. And the fourth 0px will change the LEFT side of the object.

Other possibilities:

- **height:** Just like width, this specifies the height of the object and does so in pixels like this height: 125px; (or whatever number is chosen). Change that number and you change the height of the object.
- **font-weight:** This specifies things like normal or bold. So if you wanted to make text in an object **bold**, you would say font-weight: bold; and if not, you could say normal or just leave this whole thing out.

- **text-decoration:** Let's say you wanted to make text in a certain object underlined. You could use: `text-decoration: underline;` Otherwise, you could say `text-decoration: none;` or just leave it all out.
- **float:** This is used to specify the floating position of the object. So if you say `float: left;` you are telling the object to sit as far left as possible. `float: right;` tells it to sit to the right. This is commonly used for things like a sidebar. If you have a sidebar that sits to the right of your content, you may want it to float to the right of the theme.
- **border:** This specifies whether or not there's a border and what it will look like. Let's say you are working on the sidebar code and it looks like this:

```
#sidebar {  
    float: right;  
    width: 320px;  
    margin: 0px 5px 0px 0px;  
    padding: 0px;  
    display: inline;  
}
```

Let's now say that you'd like to give the sidebar a border. You could add this line of code `border: 1px solid #000000;` anywhere within these brackets `{` and `}`. What you've stated is that you want a border that goes all

the way around the sidebar that is solid, 1px thick and black in color. If you wanted it to be thicker you could say 2px or 3px or whatever size you want. If you want the line to be dotted, dashed or double you would just replace solid with either of the three just stated. And if you wanted the line to be a different color than black, you would change #000000; to whatever color code you desire. But what if you just want lines on the left and right? You would then have to specify this by saying something like: border-left: 1px solid #000000; and then border-right: 1px solid #000000; Or if you wanted to put a line on the top and/or bottom you can do exactly the same thing, but specify either border-top or border-bottom. Here are some examples of how this plays out in the CSS code:

```
#footer {
    background: #FFFFFF;
    width: 960px;
    height: 30px;
    margin: 0px;
    padding: 0px;
    text-align: center;
    border-top: 1px solid #B0B0B0;
}

.even, .alt {
    border-top: 1px solid #DDDDDD;
    border-bottom: 1px solid #DDDDDD;
    margin: 0px;
    padding: 0px;
}
```

- **text-align:** This specifies the alignment of the text (obviously :-)).

So if you wanted the text in the specified object to align to the right

you would say text-align: right; or say you wanted to center the text.

Just say text-align: center;

This covers most of the styles you would add or adjust to achieve your desired look. There are a few more, but these are the most common.

Editing PHP files:

This is something you will do a lot less than the CSS files, but just in case you DO get your fingers wet in this area, here's some basic food for thought.

The PHP files are where you would actually HOUSE the objects that the style sheet is decorating. So when you are working on the #header in your style.css file, you are adjusting the look of the header object that is called header.php. And inside this header.php file (for example) you can specify other objects to show up as well. A common object to reside in the header.php file would be the navbar. It may look something like this:

```
<div id="navbar">
```

So when you add/remove or just edit content in your PHP files, you will be doing so by targeting the object in question (the navbar, for example) and

then making your changes there. This example would make adjustments to your navigation bar.

That's as specific as I'm going to get in this handbook. We'll touch on more down the road, as we get more advanced with our frugal theme.

Caution about making changes:

I mentioned earlier how you need to be careful when making one adjustment to certain attributes because it can affect other parts of the theme. This can occur when doing things like adjusting the body width. If you were to drop the frugal theme from 960px wide to 900px wide you would need to also adjust the header width, navbar width, content width, and so on. The point is, though most things can be adjusted with a few taps of the keyboard, some are more involved. So we're going to only focus on those customizations that are on the simple side, and get into the more advanced stuff down the road. Also, it's a good idea to make your changes one at a time. Just make the change, save your changes in Wordpress and then refresh your browser to see what happens. This way, if it doesn't do what you expect or maybe you'd like to adjust it a bit more, you know EXACTLY what you need to change. Make too many adjustments at one time and you might get lost and make a mess of things (trust me, it happens!). Also, if you make a change and nothing happens, you may want

to clear your browsers cache and then refresh the page again. If you still see no change, then you need to go back to the file you're editing and double check to make sure you make the proper adjustment.

Another precaution to make is to backup your theme files after making your desired changes. This way, if you mess something up down the road, you can just upload the backed up file/s and get things back to a pre-messed up position.

Breaking down the style.css file for easy customization:

To make things super simple, I've broken down the sections of your frugal theme style.css file by name and number. So if you want to adjust a certain part of your theme, just use this as a key, to find exactly where and what to adjust. Here are the sections:

- **(#1) Main Frame of Theme:** This is where you can adjust the body and the wrap. If you want to change the background color of your theme, just change the color code in the background section of the body. The wrap, on the other hand, is where you can specify if you want a border and if so, what kind, as well as the margin and padding of your entire theme.
- **(#2) Hyperlinks:** This is where you can adjust the colors of the hyperlinks on your site. visited means the color of the hyperlink when

the mouse is NOT hovering over it and hover specifies the color of the hyperlink when the mouse IS hovering over it. So just change the color codes and then see what your hyperlinks look like. It's a simple and easy method to lightly changing the color scheme of your theme.

- **(#3) Header:** Here you can adjust the header background by changing the color code next to background: (in the #header section), as well as adjust the size. You'll notice that there's a bunch of header objects to adjust here. Some you will never touch, while others you might adjust a lot until you get your perfect look. I'm not going to touch on every single piece of code in the style.css file, but I will point out the important ones for basic customization. So in the header section, you'll notice .headerleft. This is the adjustments for the title or logo of your blog. .headerright, on the other hand, will adjust the 468x60 header widget (if you use the widget).
- **(#4) Navbar:** Just like the header, you can adjust the background color as well as the text size, color and position. You'll also notice the borders. As in any CSS object, feel free to add, remove or adjust a border in the specified object. #navbarleft and #navbarright will adjust the attributes to the left side of the navbar (where your pages are displayed) and to the right side (where the subscribe section is located). Also, you can make adjustments to the #nav li a, #nav li a:link, #nav li a:visited { section (as well as the hover version of this

and the other versions below) to change the attributes of your navbars page links. So this allows you to make changes to the way your navbar pages are displayed and what they look like when you hover the mouse of them and what the sub-pages look like and so on

- **(#5) Content:** Here you can change the attributes of the entire content section (`#content`), the post and page section (`#contentmain`), the post and page area where the actual words and images reside (`#postarea`), the blockquote style (`blockquote`), as well as the 468x60 post widget style (`.postwidget`). Also, you can change the attributes of the different text styles (like `h1`, `h2`, `h3`, etc...) by adjusting things like `#content h1 {` and `#content h2 {` and so on. Don't forget, you can even change the hyperlink style here, by adjusting the visited and hover versions of these areas.
- **(#6) Images:** You don't really need to touch this, but this is where your CSS file is specifying the positioning characteristics of your theme. In other words, this is the code that help your image center when you tell it to center.
- **(#7) Sidebar:** Adjust the `#sidebar {` attributes for the entire sidebar and adjust the `#sidebar .widget {` attributes for each individual sidebar widget section. Also, if you want to change the attributes of the sidebar headers, do so by adjusting the `#sidebar h2 {` section.

- **(#8) Footer:** Here you can adjust the overall background of your footer (`#footer`) as well as the main text color and font size/style (`#footer p { }`) as well as the hyperlink properties of any hyperlink text that you may have on your footer.
- **(#9) Comments:** Adjust the `#comment { }` section for the comment box and the `#submit { }` section for the submit button that is right under the comment box. To adjust the actual comments themselves, you will need to change the `.thread-alt { .thread-even { .depth-1 { }` and `.even, .alt { }` sections accordingly. The alt and even parts are just specifying the fact that every other comment has a different style. This can be changed as well if you prefer.

Chapter 5: Custom Logo, Avatar, Favicon and Positioning

2 options per version.

Each version (frugal Theme or frugal Theme Lite) comes with 2 options and therefore 2 folders. One is a 'Custom Logo' version and the other is a 'Non-Custom Logo' version.

The 'Custom Logo' version includes the right code and files to incorporate

your own custom logo. You can either use the blank-logo.png file to create one or just create your own with your own preferred method. Then, when you've finished, save it as logo.png and overwrite the stock one in the 'images' directory of your frugal Theme. To keep things simple, be sure to re-size your logo.png file to 460x115px (pixels). This size can be adjusted quite easily, but this is the stock setting.

For the 'Non-Custom Logo' version, you need not change anything. The theme is setup to use the Wordpress title and tagline that you setup when you install Wordpress. This title and tag will be displayed in text format.

Having a custom logo can be a nice touch and a good way to visually brand your blog, but this is not a necessary requirement for successful blogging. So whatever you choose, keep your focus on your content and you'll be sure to rise above the blogging ranks in no time!

Adding a custom logo, avatar and/or favicon.

This process is super simple. If you go to the images folder that is located in your frugal theme folder, you will see a custom-avatar.jpg file. Then, in the main directory of your frugal theme you will see a favicon.ico file. Just replace any of these files with your own, custom versions and they will

appear on your blog. Just be sure to make the custom-avatar in .jpg format and name it accordingly. Also, maintain its size attributes (avatar=80x80 pixels).

Once you've done this, go to your Wordpress Dashboard and scroll down to 'Settings', then Discussion and then scroll down and select your 'Custom Avatar' and click 'Save Changes'. That should activate your custom avatar and you should then see it in the comments where the commenter does not have their own custom avatar.

For the favicon.ico, just go to this website: <http://chami.com/html-kit/services/Favicon> and upload an image that you want to make up your custom favicon, and it will change the file to a favicon.ico version. Then you can just download it and replace the favicon.ico file in your frugal directory. Be sure to use an image that is square in size. 80x80 is a good size to upload to the specified website.

Activating Threaded Comments

The frugal theme is setup to take advantage of the threaded comments feature of Wordpress, but you must activate this feature in your Wordpress Dashboard before it will show up in your theme. To do this, just go to your

Wordpress Dashboard and scroll down to 'Settings', then Discussion and then select 'Enable threaded (nested) comments 5 levels deep.' You can also change the 5 to however deep you want the comment replies to go. After doing this, just click on 'Save Changes' and you're done. Now go to your comments section and refresh. You should now see a 'reply' link below each comment.

Changing the size and/or position of the page/post area and the sidebar.

This is super simple and only requires the adjustment of a few pieces of code.

- **Change size:** To change the size of the content area and sidebar you just have to do a bit of math. In stock form, the .postarea is 590px wide, the #sidebar is 320px wide and the #sidebar .widget is 300px wide. If you wanted to increase the post/page area by, say, 100px, you would first change the .postarea value from 590px to 690px and then you would DECREASE both the #sidebar AND #sidebar .widget area by 100px. So the #sidebar value would drop from 320px to 220px and the #sidebar .widget value would drop from 300px to 200px. And if you wanted to increase the sidebar, you would just do the opposite. You would increase both the #sidebar and #sidebar

.widget by the amount and the DECREASE the .postarea value by the same amount. Just be sure that whatever you do you stick to the same total value of the sidebars and the post/page area.

- **Change position:** So let's say you'd like to position the sidebar to the RIGHT of the post/page area. It's quite simple. You just adjust 2 values. You would change the float value of .postarea from left to right and then you would change the #sidebar float value from right to left. That's IT! (In this instance you can leave the #sidebar .widget value alone because no matter what, it will stay INSIDE the #sidebar area, no matter what side it sits on.)

Final thoughts:

This is a lot to take in for the beginner and you can easily become overwhelmed if you're not careful. I find that the best way to learn this stuff is primarily through trial and error. Hopefully this handbook will give you some needed guidance and make the process more productive and enjoyable, but just make one change at a time and you will learn this stuff in no time.

There are many other things you can do to customize your frugal Wordpress theme, like adding images for example, but for now, this is a great starting

point. I hope you enjoyed this learning process and I'm looking forward to hearing about your frugal theme experience. Be sure to regularly stop by frugaltheme.com as I will be updating information and teaching more advanced customization techniques to help you get the most out of your frugal theme. Feel free to contact me if you have any questions.

Thanks and ENJOY! *Eric Hamm*