

Centering a Web Page

A lot of people use a centered, single-cell table to contain their Web pages. This has the advantage of making the text easier to read. On the negative side, this is not what the HTML table was intended to do.

It's pretty easy to center a page with a little bit of CSS. It is suggested that you make a separate CSS document called **center.css**. This prevents confusion with other CSS rules.

Here's the content of **center.css**. This set of rules can either be typed into the code by hand, or created with Dreamweaver. Comments in this example appear in **blue**:

```
body {
    margin:50px 0px; padding:0px;
    text-align:center;
}
#Content {
    width:625px;          /* set width for content box */
    margin:0px auto;     /* Right and left margins set to "auto" */
    text-align:left;     /* Counteract to IE5/Win Hack */
}
```

The external style sheet must be linked or imported to make it take effect on the Web page.

Within the HTML document, create a **<div>** named Content, which contains everything in the **<body>** of the Web page. An example follows:

```
<style type="text/css" media="screen">
    @import "center.css";
</style>

</head>

<body>

<!-- THIS DIV CENTERS THE PAGE CONTENTS -->
<div id="Content">

    The contents of the page go here

</div>
</body>
</html>
```

Floating Images!

Tables have often been used to encapsulate images and captions. While this works well, it's a lot of work if you have several images, and it's not what HTML tables were meant to be used for.

A fairly easy fix is to make two new styles to float images and their captions with much less work. We'll encapsulate each image and caption in a ``.

Warning:

This method requires short captions. The caption should not be wider than the image.

1. Create the following styles:

```
.imageleft {
    font-family: verdana, helvetica, sans-serif;
    font-size: .7em;      /* Text size for captions */
    color: #003399;      /* Sets color of captions */
    margin-right: 5px;    /* Keeps text 5px away from image */
    float: left;         /* Moves whole thing to the left */
    text-align: center;
}
.imageright {
    font-family: verdana, helvetica, sans-serif;
    font-size: .7em;
    color: #003399;
    margin-left: 2px;
    float: right;
    text-align: center;
}
```

2. Insert the image.
3. Place your cursor immediately to the right of the image.
4. Press **[Shift] [Enter]** to insert a line break.
5. Highlight the image and caption word(s).
6. Select the appropriate style (imageleft or imageright) in the Properties.

The Holy Grail: CSS Positioning

One of the most powerful features of CSS is that it can be used to determine the position of various page elements. This has multiple benefits: first, it allows us to get away from the use of tables for positioning page elements. Because we can use a single, central file to position the page elements in our site, we save a lot of time that would have been spent creating tables and tinkering with them.

Another very important benefit of CSS positioning is that allows us to disconnect the order of content in the HTML code from its position in the browser window. In other words, the main content of the page can be at the top of the HTML, with the navigation menus, banners, etc. at the bottom. Visitors with visual disabilities often use a program that reads the page content to them, in the order that it appears in the HTML code. With CSS positioning, the program can start reading with the main content, instead of having to wade through a banner description, navigation menu, etc.

We'll use some basic positioning techniques with the Hamer website (provided by the instructor). We'll move the navigation menu over to the side and restrict the size of the main content area for easier reading. We'll also put the banner image at the bottom of the HTML code to make the page more accessible to page reading software, while displaying the banner at the top of the browser window.

We will create the new styles in a separate positioning style sheet. This helps to avoid confusion with the other styles in use in the site.



Note: Look for the section markers!

The three sections of each page -- Header, Navigation Menu, and Main Content - - are all marked clearly within the HTML code, using non-displaying comment lines. This should help you when we're moving the sections around in the Web pages.

CSS Positioning in Three (Fairly) Easy Steps

Most CSS positioning projects consist of three parts:

1. Write the rules for positioning. Each selector begins with the # sign.
e.g.,

```
#NavigationMenu {  
    position: absolute;  
    top: 129px;  
    left: 3px;  
    width: 135px;  
}
```
2. **Import** or **Link** the CSS document in the **<head>** of the HTML page.
e.g.,

```
<style type="text/css" media="screen">  
    @import "position.css";  
</style>
```
3. Identify the sections in the HTML code with **<div>** and **</div>** tags.
e.g.,

```
<div id="NavigationMenu">  
    .  
    .  
    .  
</div>
```

Make a Positioning Stylesheet File

1. Open Notepad.
2. Click **File > Save As** and save the file in the same folder as the Web pages. Give it the filename: **position.css**

Background Color and Images

We'll begin by creating the rules for the background of the page. CSS allows a great deal of control over the behavior of background images. The following rules specify a white background for the site, identify a background image, and then tell the browser to repeat it down the left side of the screen (along the y-axis).

1. Open **position.css** with Notepad.
2. Add the following styles:

```
body {  
    background-color: white;  
    background-image: url(images/gray_bg.gif);  
    background-repeat: repeat-y;  
}
```

3. **Save position.css** and leave it open in Notepad.

Page Heading (Banner Image)

We'll continue by positioning the 650x126 pixel banner. We'll be using ***absolute positioning***, to specify exactly where we want the banner image to be placed. This is simpler than using ***relative*** or ***static*** CSS positioning, so it's a good choice for this first project.

The position of the header is referenced from the top-left corner of the screen. We will specify the width of the header, but not the height. That will be determined by the banner image (126 pixels), so we'll set the header height to ***auto***.

Our styles for the header begin with a comment line, to help us to identify the page heading part of the style sheet. Comments are always a great idea when style sheets become long and complex.

1. Add the following styles to the position.css style sheet:

```
/*Header, sized for 650 pixel banner image*/
#Header {
    position: absolute;
    top: 3px;      /*3px from top of screen*/
    left: 3px;     /*3px from left edge of screen*/
    width: 650px; /*Width of banner image*/
    height: auto; /*Height determined by image*/
}
```

2. Save **position.css** and leave it open.

Navigation Menu Styles

Next we design and position the navigation bar, along the left side of the screen, on top of the **gray_bg.gif** background image. Like the header, we'll be using ***absolute positioning***, to specify the menu's exact position.

The position of the menu is referenced from the top-left corner of the screen. We will specify the width of the menu, but not the height. That will be determined by the contents of the page. Additionally, we'll set a line-height of 220%. This will automatically create "double-spacing" in our menu.

1. Add the following rules to the position.css style sheet:

```

/*Left side nav bar, positioned absolutely*/
/*# at beginning is for use with a div to position menu*/
#Navigation {
    position: absolute;
    top: 129px; /*Sets top below 126px banner, +3 px*/
    left: 3px; /*Menu is 3px from left edge of screen*/
    width: 135px; /*Width of navigation bar*/
    height: auto; /*Height determined by page contents*/
    padding: 3px 0px 0px 3px; /*Padding at top and left*/
    font-family: verdana, arial, helvetica, sans-serif;
    font-size: 16px;
    font-weight: bold;
    line-height: 220%; /*Sets spacing between menu lines*/
}

```

2. Save position.css and leave it open.

Main Content Styles

Like the other two sections, the main content will be *absolutely positioned*. The tricky part is remembering how much width and height will be taken up by the header and navigation bar.

1. Add the following rules to the position.css style sheet:

```

/*Main contents of Web page*/
#Main {
    position: absolute;
    top: 129px; /*Sets top below 126px banner, +3 px*/
    left: 138px; /*135 nav menu + 3px*/
    width: 505px;
    height: auto;
    padding: 5px 0px 0px 5px; /*Padding at top and left*/
    /*Total width is 3+135+5+505=648 pixels*/
}

```

2. Save position.css and leave it open.

Activate the Positioning Styles

1. Open aboutus.htm with Notepad.
2. Type the following anywhere between the `<head>` and `</head>` tags.

```

<style type="text/css" media="screen">
    @import "position.css";
</style>

```

- Find the following part of the Web page:

```
<!-- BEGIN HEADER -->
<div align="center">

</div>
<!-- END HEADER -->
```

- Cut and paste the code shown in Step 3 to the bottom of the document, right above the `</body>` tag.
- Modify the first `<div>` tag of the text you just moved to read:

```
<div id="Header">
```

- Modify the first `<div>` tag of the navigation menu (located just below a comment stating **BEGIN NAVIGATION MENU**), to read:

```
<div id="Navigation">
```

- Modify the navigation menu, removing the | characters and replacing each with a `
` tag. When you're done, the Navigation `<div>` should look like this:

```
<div id="Navigation">
<!-- BEGINNING OF NAVIGATION MENU -->
<br>
<a href="index.htm">Home</a>
<br>
<a href="admin.htm">Administration</a>
<br>
<a href="staff.htm">Staff</a>
<br>
<a href="aboutus.htm">About Us</a>
<br>
<a href="http://www.nisd.net">NISD</a>
<!-- END OF NAVIGATION MENU -->
</div>
```

- Modify the beginning `<div>` statement that appears just below the comment stating **BEGIN MAIN CONTENT**. It should read:

```
<div id="Main">
```

- Save aboutus.htm** and **Refresh** it in your browser.
- Repeat Steps 1-9 with the `staff.htm` and `admin.htm` pages.

Modifying the Index Page

The index page was deliberately made a little harder to convert to CSS positioning, but it's still pretty easy. We have to delete all of the table information that is currently used to position the content.

1. Open **index.htm** in Notepad.
2. Move the Header to the bottom (just above **</body>**), as we did with the other pages.
3. Modify the bottom-of-page Navigation Menu section, just as was done with the other three pages. The section is marked with the comments:

```
<!-- BEGIN NAVIGATION MENU -->  
    and  
<!-- END NAVIGATION MENU -->
```

4. In the Main Content section, we need to delete all code between:

```
<!-- BEGIN MAIN CONTENT -->  
<div align="left">  
    and  
<!-- BEGIN ANNOUNCEMENTS -->
```

5. Delete the three lines between:

```
<!-- END ANNOUNCEMENTS -->  
    and  
<!-- DO NOT DELETE BELOW THIS LINE -->
```

6. Modify the three **<div>** tags of the sections for Header, Main and Navigation to read, respectively:

```
<div id="Header">  
<div id="Main">  
<div id="Navigation">
```

7. Add the following line to the embedded style sheet in index.htm:

```
@import "position.css";
```

8. **Save** index.htm and **Refresh** it in your browser.