GIMP® WEB 2.0 DESIGN

WEB 2.0 DESIGN: BACKGROUND FADES

SETTING UP THE HTML DOCUMENT

Step 1: Type the basic html code for a simple Web site with a header division, left, middle, and right content divisions, and a footer division. This the standard page layout that you have learned in your CIS131 Web Design and Development course.

```
<!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" lang="en"
xml:lang="en">
<head>
<meta http-equiv="Content-Type" content="text/html;
charset=ISO-8859-1" />
<title>Background Fades</title>
</head>
<body>
<div id = "container">
    <div id = "header">
                <p style = "text-align:center">Header
Banner Placeholder</p>
        </div><!--end header division -->
```

```
        <div id = "left_content">
        <p>Left content placeholder</p>
    </div><!-- end left division -->
        <div id = "right_content">
        <p>Right content placeholder</p>
    </div><!--end left division -->
    </div><!--end container division-->
</body>
</html>
```
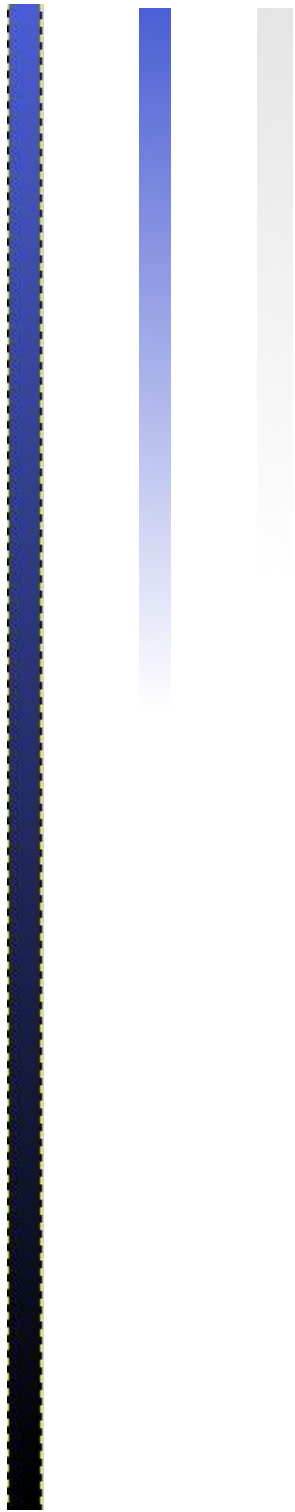
Step 2: Save the document as `background_fade.html` and view in the browser.

Header Banner Placeholder

Left content placeholder

Right content placeholder

CREATING THE GRADIENT BACKGROUNDS

Step 1: In GIMP®, create three images with blend fill. As we did with the button images, you may save the imges as 1 pixel wide and as many pixels high as you are considering your primary resolution. The sample images are displayed as 20 x 800 pixels for the body background (`body_background.png`), 20 x 400 pixels for the left_content division (`left_content_bg.png`), and 20 x 300 pixels for the right_content division (`right_content_bg.png`).

CODING THE CSS

Step 1: In an embedded style sheet or an external css style sheet, define the styles for the body and html selectors. Set the margin and padding to 0 and the height to 100%.   It is a good idea to do this on every Web page that you create.

```
body, html
    {
    margin: 0;
    padding: 0;
    height: 100%;
    }
```

At this point, you will not see a change in the browser display.

Step 2: Now you will add the blended background image to the parent element, the body. How you will create your continuous fade is to set the background **color** of the body to the same value as the last value in the background blend image. In our example, the background is black. **The background has both a color and an image value.**

```
body
    {
    /*adds a gradient fill to the body */
    background-image:url(body_background.png);
    background-repeat: repeat-x;
    /* the color is the last hex for the background
gradient */
    background-color: #000000;
    }
```

Save and test in the browser.

Step 3: The container division is required to set a height for everything except the footer, so you can "stick" the footer to the bottom of the page if the content does not reach the bottom of the page. The `height: auto !important` IS *important*. What it does is makes sure that the `height: 100` style, which comes below does not override the `height: auto` style for browsers that do not understand the `height: 100%` style.

```
#container
    {
    width: 800px;
    min-height: 100%;
    height: auto !important;
    height: 100%;
    margin: 0 auto -100px;
    background-color: white;
    }
```
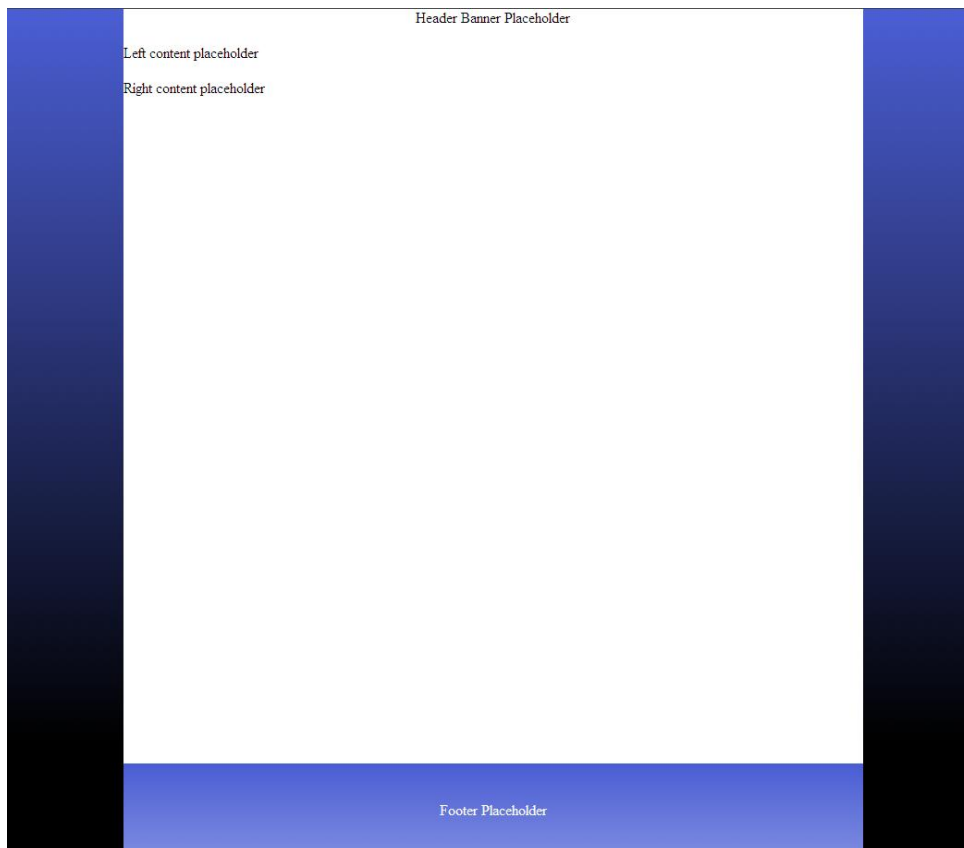
At this point, you will not see a change in the browser display.

Step 4: Style the footer with the `left_content_bg.png` (you could create a separate image for the footer background) background image, set a height in pixels that is the same as the – (negative) margin value of the #container division. A line-height with the same value as the height is required if you want to vertically

center one line text in the footer. If you have more than one line of text, you will need to experiment with the line-height value.

```
#footer
    {
    background-image: url(left_content_bg.png);
    background-repeat: repeat-x;
    height: 100px;
    width: 800px;
    margin:auto;
    text-align:center;
    line-height: 100px;
    color:white;
    }
```

Save and test in the browser.

Step 6: Sets the height of the banner to 100 and vertically centers the content in the header.

```
#header
    {
    height: 100px;
    line-height: 100px;
    }
```

Step 7: The left and right content divisions are nested inside a "content" division that is positioned between the left_content and right_content divisions.   You will create a 3-column layout by floating the left_content division on the left and the right_content divison on the right.
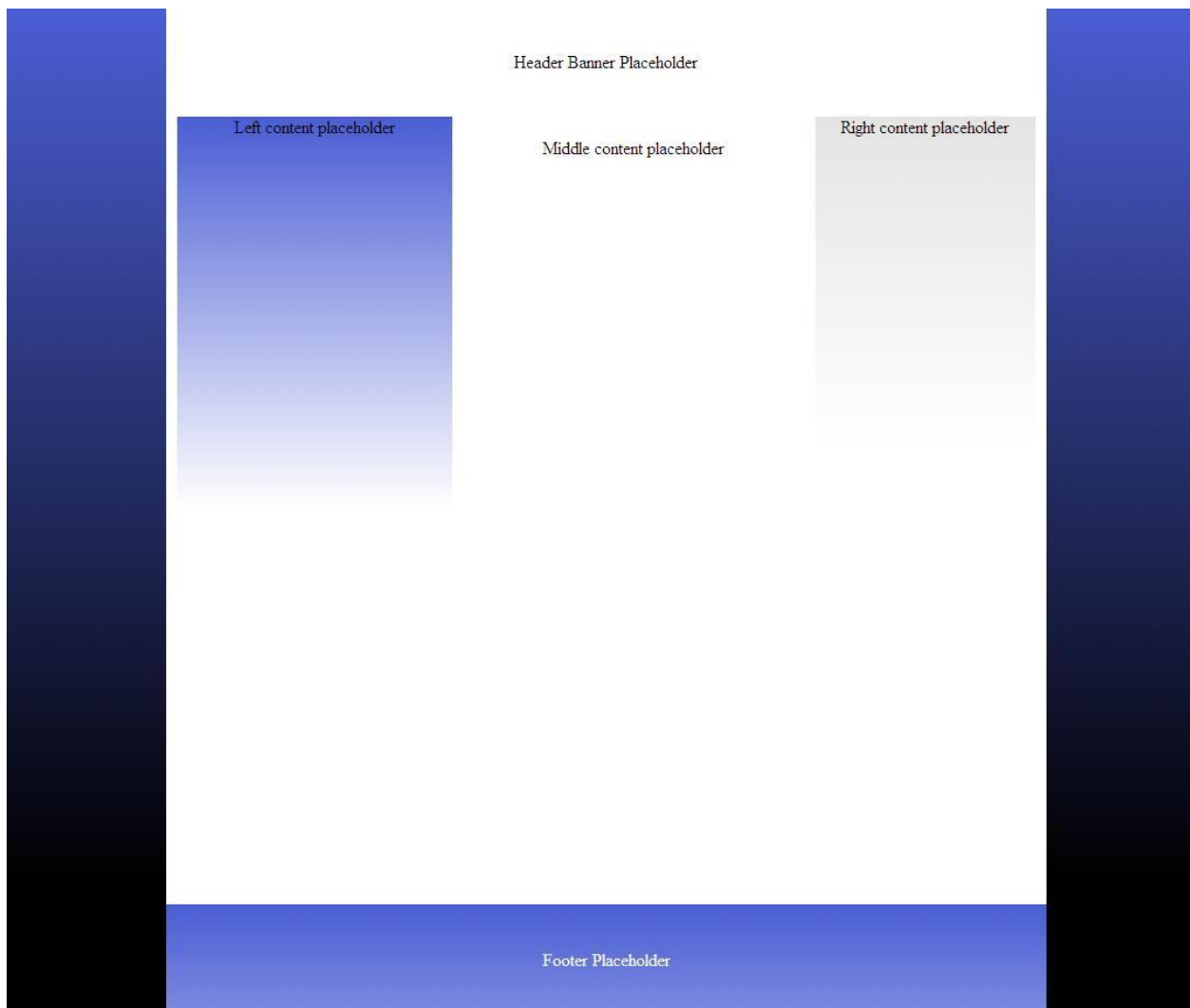
```
#left_content
    {
    width: 250px;
    height: 600px;
    float: left;
    margin-left: 10px;
    background-image:url(left_content_bg.png);
    background-repeat:  repeat-x;
    background-color:white;
    }
#right_content
    {
    width: 200px;
    height: 500px;
    float: right;
    margin-right: 10px;
    background-image:url(right_content_bg.png);
    background-repeat:  repeat-x;
    background-color:white;
    }
```

Step 8: Add a center style to the paragraph tags for visual effect as the content is limited.

```
p
    {
    text-align:center;
    }
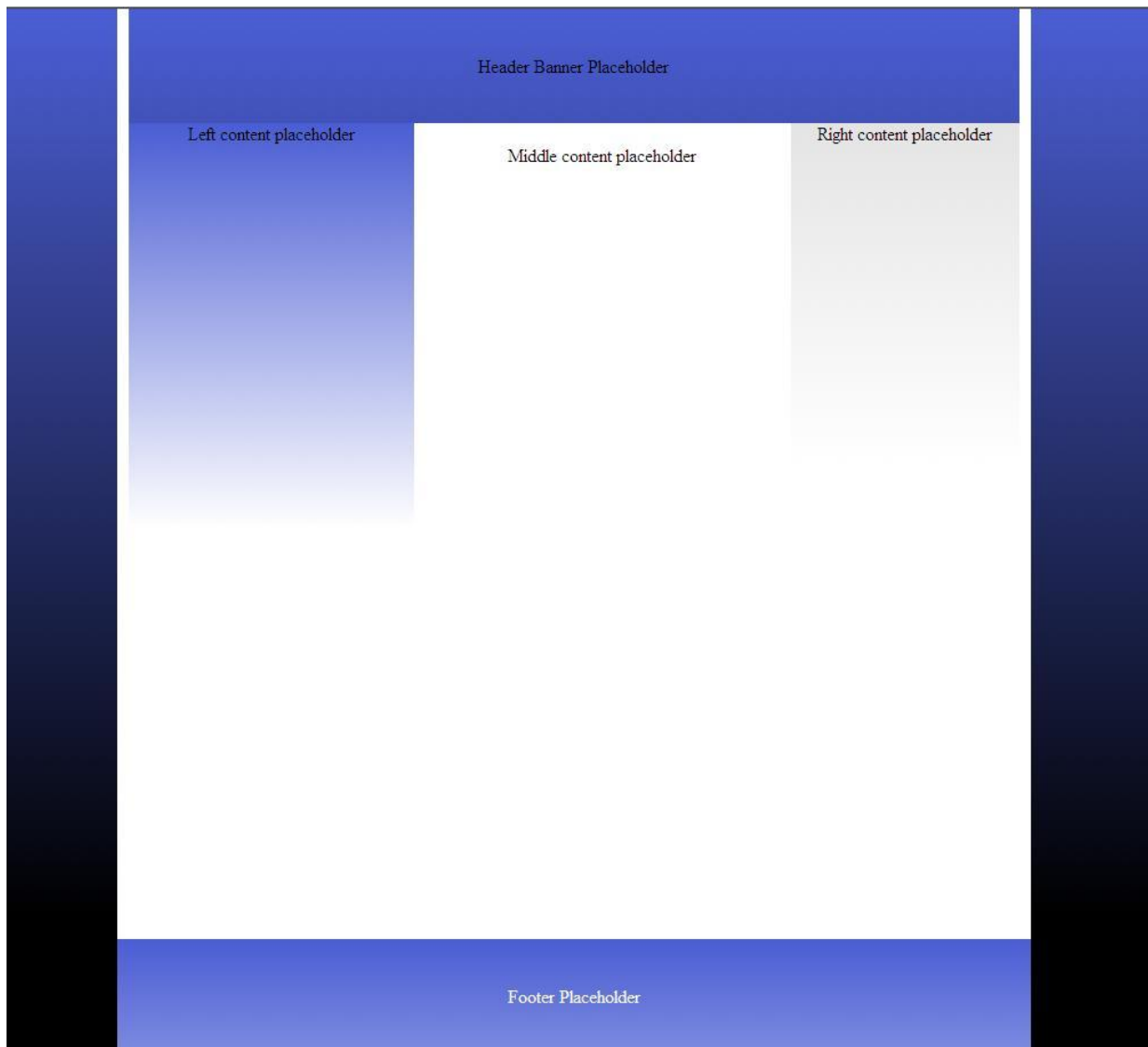```

Save and test in the browser.



**Step 9:** If the header image does not extend the complete width of the container division, you may want to add a background image to the header division. Notice that because there is a 10px left and right margin, the width of the header must be modified to 780 pixels.

```
#header
    {
    width: 780px;
    height: 100px;
    line-height: 100px;
    background-image: url(body_background.png);
    margin-left: 10px;
    margin-right: 10px;
    }
```

Save and test in the browser.

This basic example of background fades and the sticky footer should be helpful as you continue with the design exercises.