“Web Design DeCal” Hands-On Session 4 (Design)
Images

In this week’s hands-on session, we will create a playlist for a music player application. In the process, you will practice using blurred background images (faux gradient background) and building a grid of thumbnails.

Our final product will look like the demo.jpg.

I. Background

A. The background image will *remain in place* even if the user scrolls the page, and will be a *full-screen*. Remember from last week’s lecture that to accomplish this effect, you must set “width: 100%; height: 100%: position: fixed;” to #background-container.

B. It is quite difficult to use the current background image as is, since it provides terrible contrast and draws attention from the user too much. Let’s convert this image into a “gradient” by adding “-webkit-transform: scale(1.2); -webkit-filter: blur(100px);”
The first part zooms in the picture by 1.2x, and the second part simply adds a Gaussian blur of 100px. We need the first part because the Gaussian blur added by the –webkit-filter will add a white ring around it, which is an aesthetic annoyance. To avoid the ring effect and to provide better browser support, try not to use –webkit-filter and instead use Photoshop Gaussian blur for your actual project.

II. Some Typography

A. First, notice that the background is covering the content. To avoid this, we need to add a z-index to the content. Since position: static (default) is unaffected by the z-index, we need to add “position: relative” to #container in order to use z-index without changing its location.

B. Let’s set the maximum width to the content by adding “width: 840px; margin: 0px auto;” to #container. Then, add some inner spacing and contrast with “padding: 60px 0px; color: white;” to #container.

C. Title needs to be aligned center and needs more emphasis. Add “text-align: center; font-size: 36px; font-weight: 300;” to #title.

III. Grid

A. To create a grid, we will use a float’s property that it will stack things horizontally until we reach a maximum width, when it will stack things in a new line. Add “width: 180px; height: 180px; float: left;” to .item. You will see that we need some margins as well. Add “margin-left: 40px; margin-right: 40px;” to .item.

B. Notice that we now have 3 elements per row, down from 4 elements. This is because by adding margin-right: 40px to each element, each element takes up 220px of width, which is higher than the maximum width of 840px, set by the #container. To avoid this, we can take away a right margin from each element in the 4th column. To take away right margin from just the items in the 4th column – that is, every 4th div with .item class, add “margin-right: 0px;” to .item:nth-child(4n)

C. From the demo screenshot, you will see that the information about each album art is *on top of* the thumbnail. Sounds familiar? Add “position: absolute; bottom: 0px; width: 100%;” to .item-info and “position: relative” to .item in order to give reference for the position: absolute elements.

D. Let’s give the text in .item-info better contrast and spacing by adding “background: rgba(0, 0, 0, 0.5); padding: 10px;” to .item-info.

E. Notice that .item-info is overflowing. Why? It’s because we have width: 100%, which is 180px, and we add padding: 10px; on top of it, which means we have a total of 220px of space taken up by this div. To fix this, change the width of .item-info to 160px.

F. Let’s emphasize the title a bit more by adding “font-size: 16px; font-weight: 300px; margin-bottom: 5px;” to .item-title and adding “font-size: 12px; color: rgba(255, 255, 255, 0.6); ” to .item-artist.

G. For some final touch, add “box-shadow: 0px 0px 15px #4a4a4a; border-radius: 3px; overflow: hidden;” to .item.

Make sure you save your changes you made in your web inspector!!