

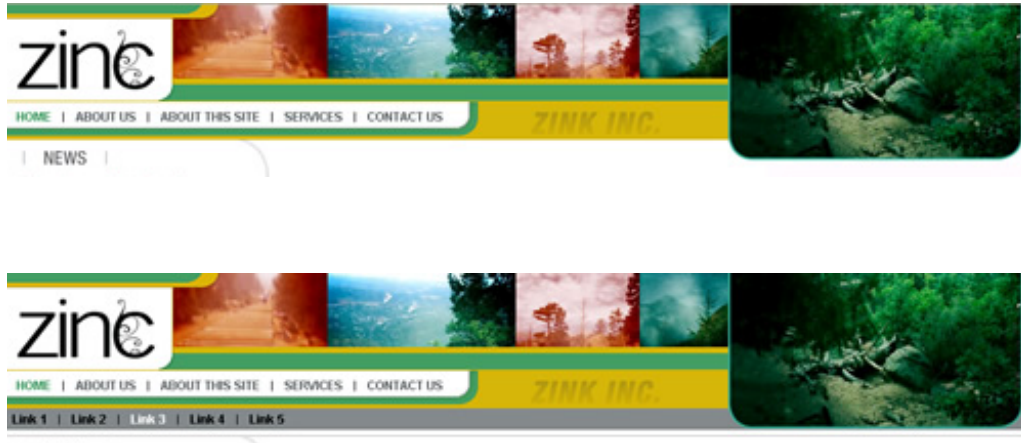
Real World HTML: Graphics-Heavy Banner

Posted At : November 11, 2010 1:00 PM | Posted By : Steve

Related Categories: HTML, CSS

In the first entry on "[Real World HTML](#)", we reviewed the design (functionally matching a site I recently completed) that we must implement and created the skeleton HTML and CSS. Now it is time to implement the header (or banner) of the design.

Let's start off by taking a look at the banner. Here is the banner in its two states:



When I looked at it, the background of the banner looked to me like a single multi-color image extending for the entire width. It would be just as valid, however, to see the background as three divs each with a different background color. The drawback to that approach is that it means that some elements in the banner have to cross div boundaries, whereas there is no need to worry about that with one solid background image. On the other hand separate divs would be able to adjust more easily to increased text size for the navigation and would allow us to make changes more easily if the client requests them. Still, this image could be changed without going back to the source document (the PSD, in this case) and this is the decision I made when I initially did this site (as much from instinct as anything else).

The next thing I want to look at is the set of four images across the top of the banner. I often see sites implemented with those as a single image. This does have advantages. It reduces the number of HTTP connections needed and the amount of HTML code to write for the images. On the other hand, it means that if the client asks us to change the order of the images or replace one with another, then we have to open up our image editing program and probably go back to the source image. This brings me to one of my most essential guidelines when working in HTML:

"Things should be as they appear."

In this case, those images look like 4 different images, so they should be. If we ever need to come back and optimize the site, then we should definitely look at this as an area of potential improvement. So long as we are able to get it to load fast enough, however, better for these each to be separate images. Keep in mind, that this guideline says "should", not "must". There will be many cases where we cannot or should not follow the guideline, but it is a good starting place.

With these images as separate images, we also need to consider the rounded curve image that extends over the left-most of the four images as well as a slight drop-shadow that comes off of the white box behind the logo. These effects demand that we make both the curved image and the white box behind the logo transparent PNGs so that they can sit on top of these images.

Let's look at our HTML so far.

```
<div id="header">
  <div id="logo"></div>
  <div id="header-pics">
```

```

    <div id="header-path"></div>
    <div id="header-plains"></div>
    <div id="header-fire"></div>
    <div id="header-mist"></div>
  </div>
  <div id="header-curve"></div>
</div>

```

Here is our CSS for the header:

```

/* HEADER */
#header {
  background-image:url(i/header-bg.gif);
  background-position:top left;
  background-repeat:repeat-x;
  position:relative;
  min-height:132px;
}
#logo {
  background-image:url(i/logo-bg.png);
  background-position:top left;
  background-repeat:no-repeat;
  float:left;
  width:162px;
  height:95px;
}
#header-pics div {
  background-position: top left;
  background-repeat: no-repeat;
  float:left;
  height:72px;
}
#header-path {background-image:url(i/headerpic-path.jpg);width:151px;}
#header-plains {background-image:url(i/headerpic-plains.jpg);width:184px;}
#header-fire {background-image:url(i/headerpic-fire.jpg);width:121px;}
#header-mist {background-image:url(i/headerpic-mist.jpg);width:87px;}
#header-curve {
  background-image:url(i/header-curve.png);
  background-position:top left;
  background-repeat:no-repeat;
  position:absolute;
  top:0;
  left:0;
  width:207px;
  height:14px;
}

```

I put my images in an "i" folder, but that is just personal preference (admittedly guided, in part, by a desire to save characters for each image reference).

The main thing that you may notice is that so far I haven't put any images in my HTML. I have done them all as background images in the CSS. This does add more code overall than using IMG tags. However, it creates more code in the CSS file (which will be downloaded only once per visit) and less in the HTML (which will be downloaded on every page). Moreover, I think of IMG tags as content. Images that are part of the design and not content at all should be background images. This also means that as we go, the print view of our site should be very close to correct even without a print style sheet. So, now another guideline:

"Design image with no ALT attribute should be background images."

This brings me to the logo. The logo for our client (unlike the white area behind it is part of the design, but it is also content. It should have an ALT attribute, so it should be an image. I chose a transparent GIF for the image. It will allow it to sit on top of any color without the extra file size of a transparent PNG which probably isn't necessary for this image.

Here is out logo div now:

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

The logo image itself brings me in mind of another guideline:

"Always crop images as closely as possible."

We shouldn't have any white space around the logo in the image itself. We can do that with CSS. This will make it easy to change positioning as we need without any need to edit the image. I can't tell you how many times I have run into trouble getting elements to align as desired because the image had white space in it.

Here is the CSS for the logo div:

```
#logo {
  background-image:url(i/logo-bg.png);
  background-position:top left;
  background-repeat:no-repeat;
  float:left;
  width:162px;
  height:95px;
}
#logo img {
  padding:25px 0 0 10px;
  z-index:2;
}
```

So the only things left for our header (other than navigation, which I will cover next time) are the large image on the right and the "ZINK INK" text. Since I can't quite duplicate that text formatting that will be an image.

In following the "Things should be as they appear" mantra, each of these should be one image. Noting that the area underneath the bottom of the large image on the right can be different colors depending on the existence of navigation that isn't on every page, this will have to be a transparent PNG. This one actually gives me a bit of pause. To do such a large photographic image as a PNG makes for a very large file size. It is 93K, which is a bit larger than I prefer for all of the client-size code (HTML/Images/CSS/JS) combined. In fact, all of the other images on this site add up to only 50.7K. So this is a very large image.

I'm going to leave that as is for the time being, but if the site needs optimization, that will be the first place that I look.

I'll make the other image a transparent PNG as well, because it is still just 2K so I may as well give myself some flexibility for later (if the client decides to change the color behind it, for example).

I will give both of the images absolute positioning from the top right of the header div itself. Absolute positioning calculates from the nearest ancestor that is absolutely or relatively positioned. This is why a used relative positioning on the "header" div even though I didn't move it at all.

So, here is the complete HTML for the header:

```
<div id="header">
  <div id="logo"></div>
  <div id="header-pics">
```

```

    <div id="header-path"></div>
    <div id="header-plains"></div>
    <div id="header-fire"></div>
    <div id="header-mist"></div>
  </div>
  <div id="header-zink"></div>
  <div id="header-image"></div>
  <div id="header-curve"></div>
</div>

```

Here is the CSS:

```

/* HEADER */
#header {
  background-image:url(i/header-bg.gif);
  background-position:top left;
  background-repeat:repeat-x;
  position:relative;
  min-height:132px;
}
#logo {
  background-image:url(i/logo-bg.png);
  background-position:top left;
  background-repeat:no-repeat;
  float:left;
  width:162px;
  height:95px;
}
#logo img {
  padding:25px 0 0 10px;
  z-index:2;
}
#header-pics div {
  background-position: top left;
  background-repeat: no-repeat;
  float:left;
  height:72px;
}
#header-path {background-image:url(i/headerpic-path.jpg);width:151px;}
#header-plains {background-image:url(i/headerpic-plains.jpg);width:184px;}
#header-fire {background-image:url(i/headerpic-fire.jpg);width:121px;}
#header-mist {background-image:url(i/headerpic-mist.jpg);width:87px;}
#header-zink {
  background-image:url(i/zink-inc.png);
  background-position:top left;
  background-repeat:no-repeat;
  position:absolute;
  top:104px;
  /*left:462px;*/
  right:362px;
  width:127px;
  height:24px;
}
#header-image {
  background-image:url(i/header-pic.png);
  background-position:top left;
  background-repeat:no-repeat;
  position:absolute;
  top:0;
}

```

```

    right:7px;
    width:289px;
    height:152px;
}
#header-curve {
    background-image:url(i/header-curve.png);
    background-position:top left;
    background-repeat:no-repeat;
    position:absolute;
    top:0;
    left:0;
    width:207px;
    height:14px;
}

```

One thing I'll note here that I didn't before is the use of the cascade - you know the first "C" in "CSS". I feel it is underappreciated by many. One great example is the set of four images at the top of the header. I put them in a div with an id of "header-pics". I knew I wanted them each top have a top-right aligned background image that didn't repeat and that they would each be 72 pixels tall and left floated. So, I defined that all in one place.

```

#header-pics div {
    background-position: top left;
    background-repeat: no-repeat;
    float:left;
    height:72px;
}

```

This isn't the same as defining a variable, it is taking advantage of a cascade. Too many people, I think, try to treat CSS as a programming language. I suspect it is because of the curly braces. They are just a handy way to group things together. It doesn't mean CSS was ever intended to be, or should ever be viewed as, a programming language. It isn't. That is OK.

In any event, now I don't have to define those things for each of the images. I just need to define which background image to use and the width of the div.

That is it. I am sure I didn't cover this as well I should have, so help me out and let me know of anything I should have covered more clearly.

Next up will be the navigation. So, take a look at the navigation and start thinking about how you would handle it. Next week I will cover how I did it.