

Golden Mean

Douglas Bowman, page designer

I returned last night from a relaxing holiday weekend spent in Pismo Beach with a few good friends. One afternoon, while most of the others retired to their rooms for a midday siesta, I pulled out my laptop, settled into a big rocking chair looking out into a garden filled with native California shrubs, trees, flowers, and herbs, and was finally able to polish off a design submission for the **CSS Zen Garden**.

[*Golden Mean*](#) is my attempt to combine colorful imagery, layered typography, and a classic three-column layout to create a simple, elegant design. There are no groundbreaking techniques used in this version, aside from those I've already used and documented on stopdesign.com. Instead, I allowed myself to focus on visual design sensibilities, playing with proportions, balance, contrast, emphasis, texture, and composition.

The title implies something very precious or highly valuable (or more simply, something golden in color). More importantly, the title was chosen for its traditional meaning: "the course between two extremes." In this application, it addresses the path I've chosen in finding the balance between absolute design and code, or form and function, if you will.

Once the design was 95% finalized, I began tackling the implementation, using a new style sheet created from scratch. Composing the CSS for this design posed very few problems. But the unique issues I encountered and the techniques I used are worth summarizing here:

Backgrounds

The first problem I ran up against involved the header background. Although I wanted all content centered in the browser window at a width of 730px, I wanted the backgrounds of the `#pageHeader` and `#quickSummary` divs to stretch to 100% of the browser width, exaggerating the "T" effect of the layout.

I initially thought I could just position the `#pageHeader` and `#quickSummary` divs absolutely, removing them from the normal flow and the width/margin constraints placed on the `#container` div. However, I'm using `position: relative` on the `#container` div to reset it as the new containing block for both the `#preamble` and `#linkList` divs. Since any object inside the relatively positioned `#container` falls subject to its limitations, I resorted to a simple solution of using a background image on the `body` element instead. Possible, because I knew the exact height of the header since it consisted entirely of images and other elements with heights set in pixels.

Order of Content

One of the difficulties of styling content is that you may not always have control over the *order of content* in the original markup, (i.e. first the header appears, then summary, then preamble, then supporting text) or the ability or allowance to change any of the markup for convenient presentation purposes. There may also be valid reasons for keeping content in a certain order to maintain proper document hierarchy, regardless of the desired presentation. This can potentially eliminate the option of using `float` as a means to group content into side-by-side columns.

Such is the case when applying style to the Zen Garden's markup. Using `position: absolute` for a few of the divs frees us up to layout the page with pieces of content in specific positions without relying on their order within the markup. Although the preamble (*The Road to Enlightenment*) exists within the `#intro` div — which uses left and right borders to frame itself — we can position the `#preamble` div absolutely to a spot relative to the right edge of the `#container` div, completely outside and below the bottom of the `#intro` div.

Similarly, the second paragraph inside `#quickSummary` (*"Download the sample..."*) can be repositioned on top of the `h2` element (*"The Beauty of CSS Design"*) by using an absolute position relative to the top-right corner of the `#container` div.

Double-element Lists

The “*Select a Design*” list is tricky to style if you’d like to separate the elements of each list item. Although the list may seem appropriately marked up as an unordered list, each list item contains both the **title** link and the **creator** link, with the word “by” sitting in between each link, as follows:

```
<li><a href="file">Golden Mean</a> by <a href="file" class="c">Douglas  
Bowman</a></li>
```

The structure above is fine if title, “by”, and creator are intended to fall together on one line (as most of the submissions so far have allowed). But I wanted more separation between the two than just font styling and color changes. Specifically, just as Josh Ambrutis styled the design list in [his version](#), I think the list looks better when the submission title appears on its own line (or set of lines, depending on container width), then the “by *creatorname*” gets forced to a new line below the title.

It seems a structure like this might actually be better suited for a **definition list**, rather than an *unordered list*. This way, the term (<dt>) could be the title, and the definition (<dd>) for each term could be the creator name, allowing each piece to display as block-level elements by default, or as inline elements with the help of the `display` property.

But alas, an unordered list is in use, and we’re left with the task of styling this list to an acceptable format. Fortunately for us, Dave left a small hook in the markup which can be used to achieve the definition list effect mentioned. As can be seen in the abbreviated list item markup I included above, the creator link is differentiated from the title link by a simple `class="c"` which gets consistently applied to the creator link element. This allows us to set **all** links within the list to block-level elements, then override the `display` property (with a bit more specificity in the selector) for the links which possess a class of “c” by setting those links back to inline-level elements, as follows:

```
#lselect a {  
  display:block;  
}  
#lselect a.c {  
  display:inline;  
}
```

Styling the links in this manner forces the title link to become a block-level element, while the creator link remains an inline-level element and appears beside the word “by” as I originally desired.

A note of caution when setting links to block-level elements with `display:block`: Most modern browsers will extend the width of the link element to fill the available width of its parent element. While the link underline (if left on) may not extend as wide as the parent element, the clickable area of the link usually will. This could potentially cause overlap of content if the `display` property is used carelessly.

Design Process To Come

As has been the case for others who submitted prior designs, creating this design for the Garden was a stimulating exercise which provided a great outlet for creative expression. The experience was rewarding from start to finish. Although the project is completely independent, and could have been slammed together in a day of work, I found myself taking a more drawn out traditional approach with the process. I started documenting each of the steps I took, initially for my own benefit. I realize it may be useful to share a condensed summary of the design process I used for my Garden submission here, so I’ll be refining the documentation and posting it here within the next couple of days.