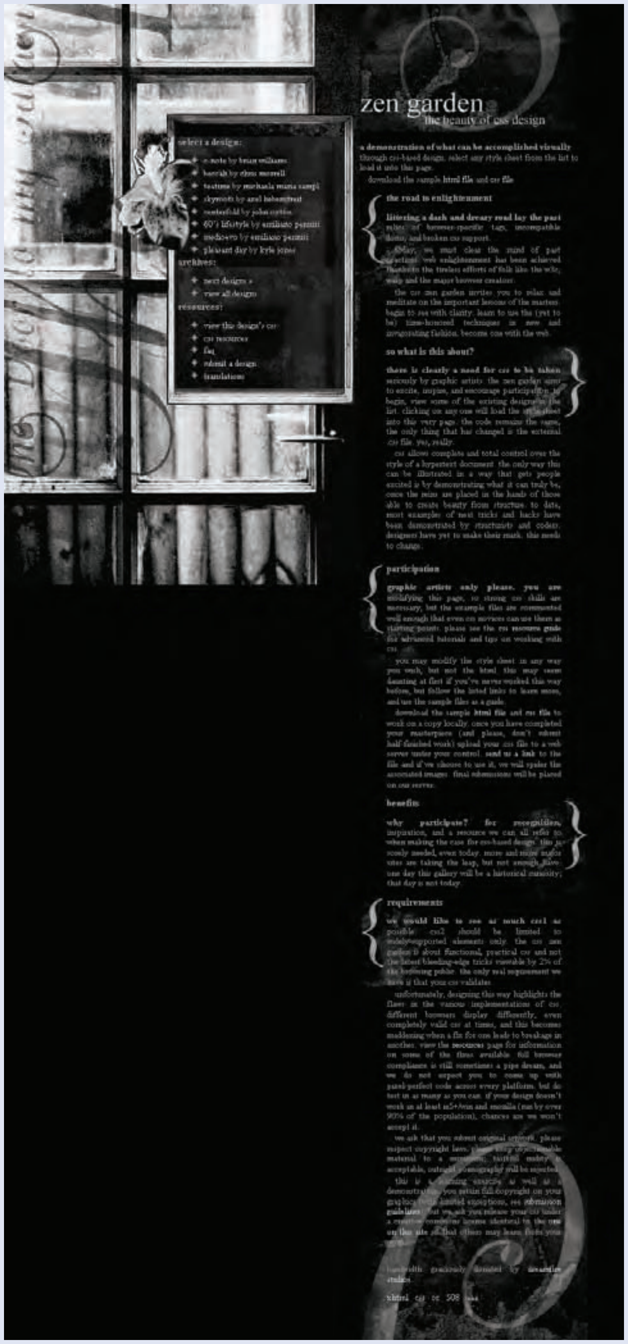


Patrick H. Lauke, Designer

www.csszengarden.com/041

door to my garden

Simple CSS resulting in impressive, sophisticated visual effects

AN AVID PHOTOGRAPHER, **Patrick Lauke** prefers to take his own photographs rather than rely on stock images. His Zen Garden entry **door to my garden** is no exception. Lauke took the photograph used in the stunning body background.

Along with photography, Lauke is fascinated by the *incidental design* concept, also referred to as *found art*—the use of found pieces in artistic endeavors. In the case of door to my garden, Lauke scanned rusted metal pieces that he stumbled across just outside of the building where he works. Along with his photographic skills, these found pieces added to the grungy, faded feel of the design.

Background Savvy

While the imagery and typography used for door to my garden certainly go a long way toward creating the dark, brooding mood that the work projects, the effects largely focus on the use of background graphics. Of course, this could be said for many of the Zen Garden designs, but door to my garden illustrates very clearly how positioning backgrounds and defining their scrolling behavior can create a unique look.

The main effect is fashioned rather simply and works well across various browsers. Lauke placed a background into the body and fixed it there (**FIGURE 1**). Then he created and set his content area to the right and overlapping the background image, allowing the content to scroll above the fixed background. Working with background images is a cornerstone of CSS design. It's essential that you understand the properties available and how they can affect and influence the way your elements are styled.

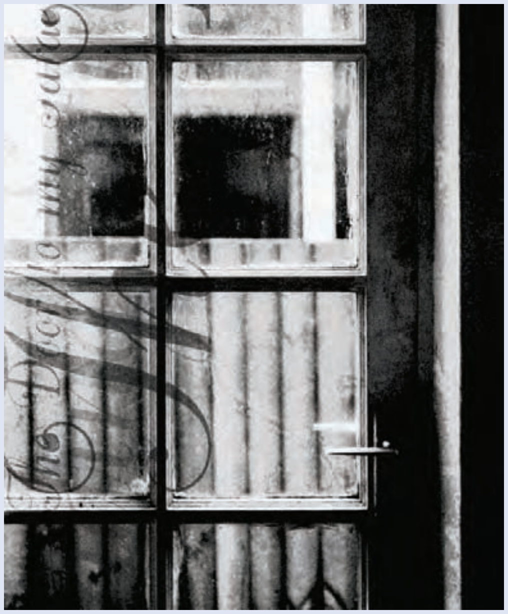


FIGURE 1 **The background graphic for the** `body` **element.**

BACKGROUND PROPERTIES IN CSS

There are six background properties, including the `background` shorthand property. **TABLE 1** shows the properties and what they do.

TABLE 1 **Background Properties in CSS**

Property	Description
`background-color`	Defines a background color.
`background-image`	Allows you to apply an image to a background.
`background-repeat`	Defines whether your background is repeated or not, and on which axis.
`background-position`	Allows you to position the background graphic in relation to the element box.
`background-attachment`	Defines whether your graphic scrolls with the page content or not
`background`	The shorthand property that combines all of the above properties.

ADDING A BACKGROUND IMAGE AND COLOR

To add an image to a background using CSS, you would use the `background-image` property with a URL value pointing to the image's location, either relative to the CSS file on the local server, or with an absolute URL pointing to another server for cross-site styling:

```
background-image: url(images/my.jpg);
```

To add color to an image, you would use the `background-color` property with a color value (hex, hex shorthand, RGB, and supported color name values are all acceptable):

```
background-color: black;
```

You can also use the `background` shorthand property to apply both color and image together along with other properties as Lauke does:

```
body {
   background: #000 url(background.png)
   -30px 0px no-repeat fixed;
}
```

Another feature of the `background-color` property that is used frequently in door to my garden is the `transparent` value. This allows an element's background to show whatever is behind it. Transparency is used with links in this design, for example:

```
a {
   color: #eee;
   background: transparent;
   text-decoration: none;
}
```

note

Many designers define a background color as well as a background image. Colors are managed by the browser and render along with the rest of your CSS, whereas images might take longer to load. Including a color along with a background graphic can create a smoother effect for your visitors.

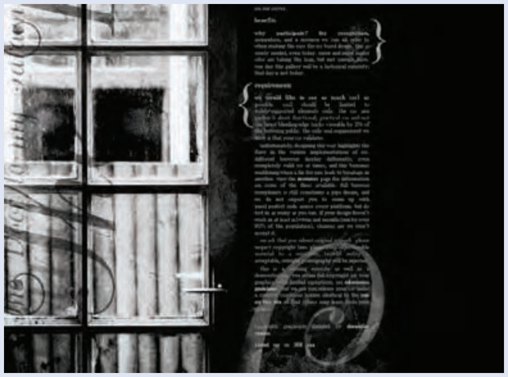


FIGURE 2 The background image remains fixed as the content scrolls. This is why the very bottom of the content and the background graphic can be seen at the same time.

ATTACHING BACKGROUNDS

The `background-attachment` property takes a value of either `scroll` or `fixed`. The default value in browsers is `scroll`, where the background image scrolls along with the content. By fixing the image, you can achieve the effect that Lauke did. The background *does not move* as the content itself scrolls (**FIGURE 2**). The photo Lauke took is placed into the `body`'s background and fixed:

```
body {
   background: #000 url(background.png) -30px 0px no-repeat fixed;
}
```

POSITIONING BACKGROUNDS

The `background-position` property controls where a background image is placed, in relation to the parent element. This property takes percentage, length, and keyword values. Each value has two potential positions. If two values are used, the first is horizontal and the second is vertical:

```
background-position: 10px 20px;
```

If only a single value is specified, it is applied to the horizontal position and the vertical position is assumed to be 50 percent or center. This declaration would place the image 10 pixels on the horizontal axis and 20 pixels along the vertical. In the background position example from Lauke's CSS, you'll notice that the positioning uses a length value of pixels:

```
body {
  background: #000 url(background.png) -30px 0px no-repeat fixed;
}
```

An item of interest is the negative value of `-30px`. This is a legitimate value for background positions and offsets the background image 30 pixels horizontally to the left, instead of to the right as positive values normally position the background. The 0-pixel value sets the vertical position of the image so that it's flush with the top margin of the containing element—in this case, the `body` element. **FIGURES 3** and **4** allow you to compare the left position of the background image, first in relation to the browser, then styled using the negative horizontal value.

CONTROLLING BACKGROUND TILING

Using presentational HTML, there is no way to control the way in which backgrounds repeat—the default is to tile them along both the *x* and *y* axes. Fortunately, CSS gives us far more control via the `background-repeat` property.

This property allows for these values: `repeat-x` so the background only tiles along the horizontal axis; `repeat-y` so the background only tiles along the vertical axis; `no-repeat` so the graphic doesn't repeat; and `repeat`, which is the default value and specifies that the graphic repeats along both axes. Lauke has used the `no-repeat` value in his background shorthand:

```
body {
  background: #000 url(background.png) -30px 0px
  no-repeat fixed;
}
```

tip

You do not need to use a unit if you're setting a value to `0`. This is true with any CSS property. A value of `0` is implicit. So, another way to have approached this particular declaration would have been to write `background: #000 url(background.png) -30px 0 no-repeat fixed;`

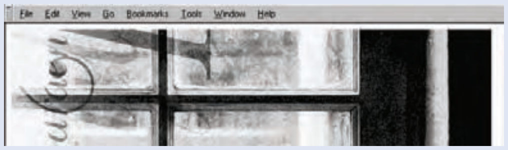


FIGURE 3 **The image in normal flow.**

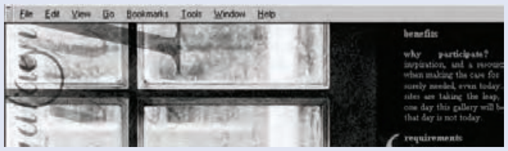
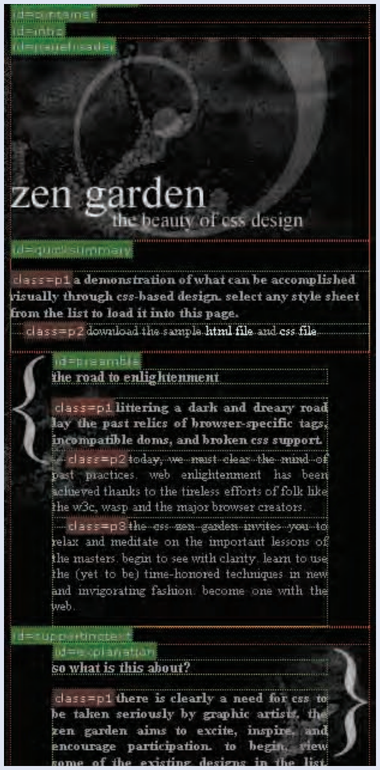


FIGURE 4 **Visualizing the negative value in the background's position.**

FIGURE 5 **The padding is what sets the container section 470 pixels from the left.**



id=container
id=intro
id=pageHeader

zen garden
the beauty of css design

id=quickSummary

class=p1 a demonstration of what can be accomplished visually through css-based design. select any style sheet from the list to load it into this page.

class=p2 download the sample html file and css file

id=preamble
the road to enlightenment

class=p1 littering a dark and dreary road lay the past relics of browser-specific tags, incompatible doms, and broken css support.

class=p2 today, we must clear the mind of past practices. web enlightenment has been achieved thanks to the tireless efforts of folk like the w3c, wasp and the major browser creators.

class=p3 the css zen garden invites you to relax and meditate on the important lessons of the masters. begin to see with clarity. learn to use the (yet to be) time-honored techniques in new and invigorating fashion. become one with the web.

id=supportingText
id=explanation
so what is this about?

class=p1 there is clearly a need for css to be taken seriously by graphic artists. the zen garden aims to excite, inspire, and encourage participation. to begin, view some of the existing designs in the list.

FIGURE 6 **Using the Web Developer extension, you can see the styles as applied to elements within the** `#container` **division.**

This ensures that the background image doesn't tile. In all backgrounds in door to my garden, there is consistently no background repetition, though other instances are controlled through further style rules not shown here.

Layout Choices

To get the 300-pixel-wide `#container`, which is offset from the left of the viewport by 470 pixels, Lauke set a 470-pixel padding into the `body`:

```
body {
   padding: 0 0 0 470px;
}
```

Remember that the order for shorthand properties in padding and margins is `top`, `right`, `bottom`, `left`. All of these values have been set to `0` (you'll note that here, Lauke chose not to include the length value for 0s) except for the `left` value, which pads the left of the canvas with 470 pixels of space (**FIGURE 5**).

The container is styled with a background image, a color, and a width only:

```
#container {
   background: #000 url(bottom_corner.png) no-repeat bottom right;
   color: inherit;
   width: 300px;
}
```

Using the Web Developer extension for Mozilla and Firefox, you can turn on the "outline block element" feature and also get the class and ID information for each element. **FIGURE 6** demonstrates the way in which styles have been applied to elements within the `#container` division.

Additional Style Effects

door to my garden relies on mostly uncomplicated and hack-free CSS, so one of the more challenging aspects Lauke faced was managing the menu and flower image that appears to accompany it (**FIGURE 7**). The framed menu was hard to get working consistently across various browsers.

The menu portion itself wasn't the real challenge. Lauke created a container for the `#linkList ID` and built the navigation by styling each element with a different piece of the background that forms the frame. But getting that flower image floating out of the framed box and over the background was more complicated.

```
#linkList2>#lselect {
   background: url(flower.png) no-repeat top left;
   margin-left: -65px;
   padding-left: 80px;
   min-height: 150px;
         }
```

Using a child selector, this CSS works in Mozilla, Opera, and Safari browsers. In order to get acceptable results in Internet Explorer, Lauke used one of the extra `divs` in the Zen Garden markup to absolutely position the flower image:

```
#extraDiv1 {
   position: absolute;
   top: 165px;
   left: 142px;
   background: url('flower.png') no-repeat top left;
   width: 115px;
   height: 150px;
   }
```

At this point, fully compliant browsers are getting both styles. So Lauke had to go one step further and hide this particular style from those browsers already supporting the original `#linkList2>lselect` style:

```
body>#extraDiv1 {
   display: none;
   }
```

Voilà! The flower and menu are visible and behave alike in modern browsers.

Great Style, Simple CSS

It's clear that the beauty of door to my garden is due to unique imagery, typography, and the thoughtful placement of elements. While at first glance the design appears complicated, the reality is that it uses simple CSS to achieve the majority of its effects, with only one true workaround to ensure compatibility.

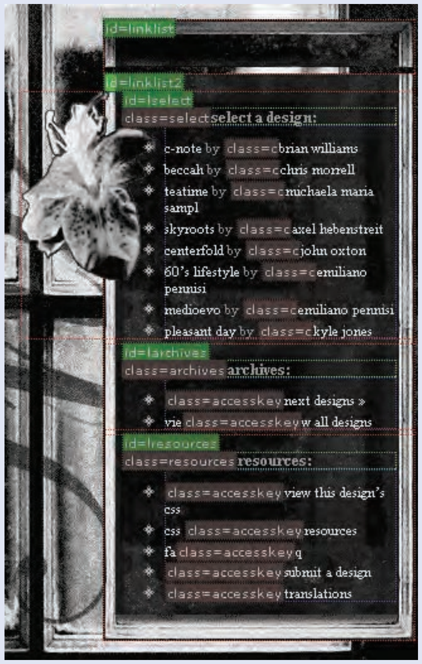

id=linklist
id=linklist2
id=lselect
class=select select a design:
c-note by class=cbrian williams
beccah by class=cchris morrell
teatime by class=cmichaela maria sampl
skyroots by class=caxel hebenstreit
centerfold by class=cjohn oxton
60's lifestyle by class=cemiliano pennisi
medioevo by class=cemiliano pennisi
pleasant day by class=ckyle jones
id=larchives
class=archives archives:
class=accesskey next designs »
vie class=accesskey w all designs
id=lresources
class=resources resources:
class=accesskey view this design's css
css class=accesskey resources
fa class=accesskey q
class=accesskey submit a design
class=accesskey translations

FIGURE 7 The menu and flower image, outlined so that you can see the elements in question.

tip

You can get the Web Developer extension free from chrispederick.com (www.chrispederick.com/work/firefox/webdeveloper/index.php).