CS7026 – CSS3

CSS3 – backgrounds and fonts
What You’ll Learn…

- The `background-size` property to scale a background image with the text
- Multiple background images on one element
- The `border-image` property to create graphic borders
- The `background-clip` property to move a background image out from under a border
- The `@font-face` rule to embed unique fonts in the page
The Base Page

- We’re going to create something that looks like it was written on a piece of real paper.

- The first step might be to apply a simple lined paper background image to it.
Scaling the Background Image

- One thing that would make the background look more realistic is if the text were aligned to the notebook paper lines, instead of overlapping them indiscriminately.

- To fix this without CSS3, you would need to set a base font-size and line-height in pixels, and then adjust the spacing between the lines in your background image to match.

- This would work for most users. But if anyone resized the text, or had non-standard user settings to override the pixel font sizes, the text would become misaligned.

- The text could scale, but the background image couldn’t
Scaling the Background Image

- With `background-size`, you can control the horizontal and vertical scaling of a background image as well as how it stretches to cover the background area and gets clipped.
How `background-size` Works

- This is an image 200px wide by 120px tall.

- This is how it looks when set as a normal repeating background of a div that’s 500px wide by 200px tall;

- Since the div’s dimensions aren’t an even multiple of the image’s dimensions, some of the image gets cut off on the right and bottom.
How `background-size` Works

- We can use the `background-size` property to scale the image down from 200 pixels to 100 pixels wide:

```css
#star {
  width: 500px;
  height: 200px;
  border: 1px solid #999;
  background-image: url(images/star.gif);
  background-size: 100px auto;
}
```
How `background-size` Works

- The first value in the `background-size` property, 100px, sets the width of the background image.

- The second value, `auto`, sets the height. A value of `auto` makes the height whatever it needs to be to preserve the aspect ratio of the image.

- If you leave the second value out, the browser assumes it to be `auto` (so a value of `background-size: 100px;` would have worked the same here).
How `background-size` Works

- So the background image has been shrunk but still keeps its aspect ratio:

![Background Image](image.png)

- The browser has scaled the image to 100 pixels wide, so it now fits in the `div` exactly five times and doesn’t get cut off on the right.
How `background-size` Works

- If you use percentages in the `background-size` property, they’re relative to the box the background is on, not to the background image itself.

- If you wanted exactly two copies of the image to show in the `div`, with neither cut off at all, you could use this CSS:
  ```css
  div {
    width: 500px;
    height: 200px;
    border: 1px solid #999;
    background-image: url(images/stars.gif);
    background-size: 50% 100%;
  }
  ```
How background-size Works

- The image is stretched to fill half the width of the div and all of its height, and then repeated.

- The browser has to both distort the shape of the image and scale it up, making the edges in the image look a little pixelated.

- As with any browser-based scaling, background sizing doesn’t look good with all images, but can work quite well with grungy, abstract, or very simple images that don’t have super-clean edges - such as our lined paper background.
More New Ways to Tile Backgrounds

- Another way to keep background image tiles from getting cut off on one or more sides is to use the new CSS3 values of `round` and `space` in the `background-repeat` property.

  - `round` repeats the background image but rescales it so it will fit an even number of times without getting cut off.

  - `space` repeats the background image as often as it will fit without getting cut off, and then spaces the tiles out to fill any leftover room.
Making the Paper Lines Scale With the Text

- In order to make our paper background image scale with the text, we need to set its dimensions not in percentages or pixels, but in **Ems**.

- As we know, **Ems** are a relative unit of measurement based on the current font height.
Find the `#paper` rule in the CSS, and add the `background-size` property, plus the Mozilla and Webkit equivalents:

```css
#paper {
  float: left;
  margin: 40px;
  padding: 3.2em 1.6em 1.6em 1.6em;
  background: url(images/paperlines.gif) #FBFBF9;
  -moz-background-size: auto 1.6em;
  -webkit-background-size: auto 1.6em;
  background-size: auto 1.6em;
}
```
Making the Paper Lines Scale With the Text

- Opera, Chrome, Safari 5, Firefox 4, and IE 9 use the standard background-size property; Firefox 3.6 and Safari 4 and earlier use the -moz- and -webkit- versions of the property, respectively.

- In each property, we’re telling the browser that we want the height of the image to be 1.6 ems and that we want the width to just size itself proportionally.

- The image depicts one line on the paper, so that means that the space between every line will now be 1.6 ems tall (the height of each line of text is 1.6, specified by the lineheight already in place on the body element:}
The text is still not lining up with the lines in the image because we haven’t set all the text sizes and margins to line up with a regular spacing of 1.6 ems.

The paragraph and list text have the correct spacing for the background image, since their line-height is already 1.6 and their bottom margins are 1.6 ems, as you’ll see in the CSS. But the headings need to have their margins tweaked to fall in line.
Making the Paper Lines Scale With the Text

h1 {
  margin: -.3em 0 .14em 0;
  color: #414141;
  font-family: Helvetica, “Helvetica Neue”, Arial,
  → sans-serif;
  font-size: 3.5em;
  font-weight: normal;
}

h2 {
  clear: left;
  color: #414141;
  margin: .5em 0 -.17em 0;
  font-family: Helvetica, “Helvetica Neue”, Arial,
  → sans-serif;
  font-size: 2.17em;
  font-weight: bold;
}
Making the Paper Lines Scale With the Text

- These margin values are based on trial and error.

- Unlike with absolute pixel-based measurements, you’re not going to be able to find values that work perfectly for all browsers; each browser has different ways of rounding and translating relative measurements like ems into the pixels displayed on the screen.

- These margin values work well for Firefox, Safari, and Chrome. Everything is spaced out at regular intervals of 1.6 ems, keeping the text aligned to the lines in the background image.
Making the Paper Lines Scale With the Text

- Once the text is aligned with the background image, if the user has a different default text size from the norm, or scales the text size up or down, the background image scales with it, keeping the lines always aligned with the text.

- Also, if you were to later change the base font size on the body element, everything would scale to match, without your having to remake the background image.
More on the **background-size** property

- The background-size property is part of the Backgrounds and Borders module, found at www.w3.org/TR/css3-background.

- Its value can be a width and height in any unit, or it can be **auto**.

- Alternately, background-size can be set to either **contain** or **cover**. Both make the browser scale the image proportionally.

- A value of **contain** scales it to the largest size where both its width and height will fit inside the background area, so it doesn’t get cut off at all, but often leaves some area with no background on it.

- A value of **cover** scales it to the smallest size where one tile of it will completely cover the background area, but allows it to get cut off where necessary to make sure the whole area has a background image covering it.
More on the **background-size** property

- Other than scaling lines to match text spacing, you might want to use background-size for:
  - Making the non-repeating background of the header of a page scale in a liquid or elastic layout to always fill the whole header width.
  - Making a repeating background image tile a full number of times instead of the tiles getting cut off on the edges of the box.
  - Making a large background image always fill the entire page; see [www.alistapart.com/articles/supersize-that-background-please](http://www.alistapart.com/articles/supersize-that-background-please).
  - Scaling a link or list item’s background image icon with its text
  - Scaling background images for high-resolution displays down by half, so that when it doubles the pixels, as it always does, the images won’t look blurry.
  - Changing the size of background images based on the size of the user’s window, using media queries.
Multiple Background Images on One Element

- Before CSS3, only one background image per box was allowed, so you’d have to add an extra div for each extra image and apply one image to each div.

- This nesting divs method wasn’t difficult, but it was messy and increased the pages’ file size.

- To add more images later, you’d need to not only change the CSS, but the HTML as well.
Multiple Background Images on One Element

- With CSS3, you can leave the HTML alone and instead simply list each background image in the background-image or background property, separated by commas.

- Each image can be positioned, repeated, sized, and otherwise controlled independently.
Multiple Background Images on One Element

#paper {
    float: left;
    margin: 40px;
    padding: 3.2em 1.6em 1.6em 1.6em;
    background: url(images/paperlines.gif) #FBFBF9;
    background: url(images/thumbtack.png),
              url(images/stains1.png),
              url(images/stains2.png),
              url(images/stains3.png),
              url(images/stains4.png),
              url(images/paperlines.gif) #FBFBF9;
    -moz-background-size: auto 1.6em;
    -webkit-background-size: auto 1.6em;
    background-size: auto 1.6em;
}

Multiple Background Images on One Element

- The first background declaration will continue to be used by browsers that don’t support multiple background images.

- Because they don’t understand the syntax of the second background declaration, they’ll ignore it.

- Browsers that do support multiple background images will override the first declaration with the second.
Multiple Background Images on One Element

- The background images are layered on top of each other, with the first declared image put on top of the stack (that’s why the thumbtack image is listed first and the lines image is listed last).

- We still need to tell the browser how we want to repeat, position, and size each image.

- To do this, treat each snippet between the commas as if it were its own standalone background shorthand property, and write each of the background-related properties in it accordingly.
Multiple Background Images on One Element

- These are all the pieces that can go in the background shorthand property. The order matters for some and not for others, so stick with the order shown:

```
background: url(top.gif) 50% 0 / 10px 100px no-repeat fixed padding-box padding-box, TOP_LAYER
url(middle.gif) 50% 0 no-repeat content-box, SECOND_LAYER
url(bottom.gif) 0 0 / auto 20px repeat #FFF; FINAL_LAYER
```
Multiple Background Images on One Element

- So...

```css
background: url(images/thumbtack.png) 50% 5px no-repeat,
url(images/stains1.png) 90% -20px no-repeat,
url(images/stains2.png) 30% 8% no-repeat,
url(images/stains3.png) 20% 50% no-repeat,
url(images/stains4.png) 40% 60% no-repeat,
url(images/paperlines.gif) #FBFBF9;
```
Multiple Background Images on One Element

- Next, modify the background-size properties to tell the browser that each image should be sized using its native dimensions, except for the last (the lines image):

  - `moz-background-size: auto, auto, auto, auto, auto, auto, auto 1.6em;`
  - `webkit-background-size: auto, auto, auto, auto, auto, auto 1.6em;`
  - `background-size: auto, auto, auto, auto, auto, auto, auto 1.6em;`

- Each comma-separated value matches up with the comma-separated value at the same spot in the background property’s value list.
More on Multiple Background Images

- Multiple background images are a new feature of the background and background-image properties, not a new property itself.

- These properties are part of the Backgrounds and Borders module, found at www.w3.org/TR/css3-background.

- List each background image in the background-image or background property, separated by commas.

- The background images are layered on top of each other, with the first declared image put on top of the stack.
More on Multiple Background Images

- Each image can be positioned, repeated, sized, and otherwise controlled independently.

- To do so, include this background styling information with each image URL in the background property, or add a comma-separated list of values to each independent background property, such as background-repeat: no-repeat, no-repeat, repeat-x, repeat.

- Each value in the list matches up with a value in the list of background images.
You might want to use multiple background images for:

- Flexible boxes with fancy or irregular corners or edges that other CSS3 properties like border-radius can’t handle, such as ornate buttons that would still need images; see http://css-tricks.com/css3-multiple-backgrounds-obsoletes-sliding-doors
- The parallax effect, where resizing a window or hovering over a div makes the images appear to move at different speeds in relation to each other; see www.paulrhayes.com/2009-04/auto-scrolling-parallax-effect-without-javascript
- Making what appears to be a single image stretch across the whole width of a box or page, while it’s really made up of multiple pieces, such as a landscape image with a sun that you always want to appear in the top right corner and a tree that you always want to appear in the bottom left corner
More on Multiple Background Images

- Distributing images across the full width or height of a box, using percentage positions to keep them spaced out from each other, such as multiple cloud images over a blue background colour.

- Creating the appearance of an object from real life, using a top image slice, repeating middle slice, and bottom slice, all on the same box.

- Applying a CSS3-generated gradient (remember, it goes in the background-image property, not background-color) along with a background image, to fade out a texture, blend the edges of an image into a solid colour, or reveal portions of an image as the user scrolls down the page; see http://atomicrobotdesign.com/blog/htmlcss/make-the-thinkgeek-background-effect-using-css3
Adding a Graphic Border

- Another graphic detail that would be nice to add is a border on the left side of the paper to make it look like it was torn from a spiral notebook.

- CSS3 allows you to assign an image to a border, in addition to (or instead of) a colour and line style.

- The browser will take a single image, slice it into pieces, and stretch or tile each of those pieces across each border.
Adding a Graphic Border

- For instance, let’s say that this is the image we want to use for the borders on a div.

- We want to use the top 30 pixels of the image for the top border, the right 25 pixels for the right border, the bottom 27 pixels for the bottom border, and the left 34 pixels for the left border.

- We need to use these values as both our border widths and our border image slice locations.
Adding a Graphic Border

```css
.clouds {
    width: 400px;
    height: 150px;
    border-style: solid;
    border-width: 30px 25px 27px 34px;
    border-image: url(clouds.png) 30 25 27 34 stretch;
}
```

- The first part of the border-image value is the path to the image, which works just like any other path in CSS.

- Next comes one or more numbers to specify where the browser should slice the image. In this case, we’re using four numbers, since we want four different amounts sliced off from each edge.
Adding a Graphic Border

- The first number, 30, is the inward offset from the top edge of the image, in pixels. The second number, 25, is the inward offset from the right edge, the third is the offset from the bottom, and the fourth is the offset from the left.

- The browser will slice the image at each of these lines, creating nine images that it applies to each border, each corner, and the middle of the box.
The Centre Slice

- The centre slice of the border image is used to cover the entire middle area of the box, inside the border area.

- The spec says that this centre slice should be discarded by default, unless you add the word `fill` to your border-image value.

- However, right now browser support for the `fill` keyword is patchy with some older ones even filling by default.

- If you want to ensure that the middle of the border image doesn’t obscure the background image or colour beneath it, use your graphics program to make the middle portion of the image you’re using transparent, and save the image as a transparent GIF or PNG.
More on the **border-image** Property

- The border-image property is part of the Backgrounds and Borders module, found at [www.w3.org/TR/css3-background](http://www.w3.org/TR/css3-background).

- It’s a shorthand property, but you can’t use the individual properties right now, since no browser supports them declared outside of the shorthand border-image property.

- In the border-image property, you specify an image, how far in from each edge you want the browser to slice the image, and how to repeat each image (except the corners) across its border.
More on the `border-image` Property

- You can use one to four slice values, depending on whether each side needs to be sliced differently.

- One value applies the same slice offset to all four sides; two values applies the first to the top and bottom and the second to the right and left; three values applies the first to the top, second to the right and left, and third to bottom; and four values applies each to an individual side, starting at the top edge and going clockwise.
More on the \texttt{border-image} Property

- The repeat value can be set to \texttt{stretch}, \texttt{repeat}, \texttt{round}, or \texttt{space}.

- Using one repeat value will apply the value to all four sides, while two repeat values applies the first value to the top and bottom borders and the second value to the left and right sides.

- \texttt{repeat} will tile all four edges plus the centre;
- \texttt{stretch} will stretch them to fill the area;
- \texttt{round} will tile and scale them so each fits a whole number of times;
- \texttt{space} will tile them so each fits a whole number of times and then evenly distribute the extra space between the tiles.
More on the `border-image` Property

- You need to always set `border-width` in conjunction with `border-image` to create a border area for the image to draw onto.

- You also need to set `border-style` to `solid`.

- There is also a `border-image-width` property, but no browser currently supports it, nor does any browser currently support `border-image-outset`.

- Sadly, border images don’t conform to curved borders created by `border-radius`.
More on the **border-image** Property

- You might want to use border-image for:
  - Gradient backgrounds
  - Scalloped edges to create the effect of a stamp or raffle ticket
  - Graphic edges to create the effect of a picture frame or certificate; see [www.norabrowndesign.com/css-experiments/border-image-frame.html](http://www.norabrowndesign.com/css-experiments/border-image-frame.html).
  - A curved or angled edge of a box
  - Box drop shadows that are curved or angled (box-shadow can do only straight drop shadows, but you can create an image of an irregular shadow and apply it as a border image)
Applying the Torn-edge Image

- Let’s put `border-image` to use in our page to apply the torn-paper edge image.

- We want to apply the image only to the left border, so we’ll make that border 50 pixels wide - the width of the image - and set the other borders to zero:
Applying the Torn-edge Image

#paper {
float: left;
margin: 40px;
padding: 3.2em 1.6em 1.6em 1.6em;
border-width: 0 0 0 50px;
border-style: solid;
background: url(images/paperlines.gif) #FBFBF9;
background: url(images/thumbtack.png) 50% 5px no-repeat,
url(images/stains1.png) 90% -20px no-repeat,
url(images/stains2.png) 30% 8% no-repeat,
url(images/stains3.png) 20% 50% no-repeat,
url(images/stains4.png) 40% 60% no-repeat,
url(images/paperlines.gif) #FBFBF9;
-moz-background-size: auto, auto, auto, auto, auto, auto, auto 1.6em;
-webkit-background-size: auto, auto, auto, auto, auto, auto, auto 1.6em;
background-size: auto, auto, auto, auto, auto, auto, auto 1.6em;
}
Applying the Torn-edge Image

Next, we’ll apply the border image, using the standard border-image property for Chrome and Opera and the prefixed properties for Firefox and Safari:

```css
-moz-border-image: url(images/edge.png) 0 0 0 50 round;
-webkit-border-image: url(images/edge.png) 0 0 0 50 round;
border-image: url(images/edge.png) 0 0 0 50 round;
```
Applying the Torn-edge Image

- We’ve set each of the slice locations to zero except for the left one; we don’t want to slice off any from the top, right, or bottom, but we do want to slice in from the left edge by 50 pixels so that the entire 50-pixel-width of the image is used for the left border.

- For the repeat value, we’ve used `round` to repeat the image but keep it from ending in the middle of a hole.

- Browsers that don’t support this value will treat it as repeat instead, which is an acceptable second choice.
Using `background-clip` to Position Images

- Our edge image is now repeating down the left side of the div, but the background image is showing through it.

- That’s because, by default, borders are drawn on top of the background area.

- Luckily, we can change this default behaviour with CSS3.
Using `background-clip` to Position Images

- CSS3 lets you control where backgrounds are placed relative to the borders with the new `background-clip` property.

- The default value, `border-box`, makes backgrounds extend under the borders as they’ve always done.

- Setting `background-clip` to `padding-box` starts the backgrounds inside the borders, under the padding area:
Using `background-clip` to Position Images

- `-moz-background-clip: padding;`
- `-webkit-background-clip: padding-box;`
- `background-clip: padding-box;`

- Chrome, Safari 5, Firefox 4, and Opera use the standard property, while Firefox 3.6 and earlier and Safari 4 and earlier use the prefixed versions.

- Note also that the `-moz-background-clip` property takes a value of `padding` instead of the standard `padding-box`. 
More on the `background-clip` Property

- The `background-clip` property is part of the Backgrounds and Borders module, found at [www.w3.org/TR/css3-background](http://www.w3.org/TR/css3-background).

- It controls under which sections of a box the background is painted.

- The allowed values are:
  - `border-box` (the default value to paint backgrounds under borders),
  - `padding-box` (to clip backgrounds at the outer edge of the padding area and not extend under borders),
  - `content-box` (to clip backgrounds at the outer edge of the content area and not extend under padding or borders).
More on the background-clip Property

- Firefox 3.6 and earlier don’t support content-box, and use values of border and padding, not border-box and padding-box; Firefox 4 doesn’t have these issues.

- Safari 5 supports the border-box and padding-box values in the standard background-clip property, but supports only contentbox in the -webkit-background-clip property.
More on the `background-clip` Property

- Other than moving a background out from under a border image, you might want to use `background-clip` for:
  - Moving a background colour or image out from under a dashed or dotted border
  - Creating the appearance of a double border, one made from the actual border and one made from the padding, by using `content-box`
  - Keeping the background colour from bleeding outside the edges of rounded corners, as sometimes happens in Webkit-based browsers, by using `padding-box`; see [http://tumble.sneak.co.nz/post/928998513/fixing-the-background-bleed](http://tumble.sneak.co.nz/post/928998513/fixing-the-background-bleed)