

Internet Programming

Web Page Graphic Design

Dan Brandon, Ph.D., PMP

Figure Boxes

- `figure` and `figcaption` elements:
 - The `figcaption` element is optional and can be placed directly before or after a figure box content

- General syntax to create a figure box:

```
<figure>  
  content  
  <figcaption>caption text</figcaption>  
</figure>
```

where `content` is the content appearing in a figure box and `caption text` is the description text that accompanies the figure

Figure Boxes (con't)

- CSS styles can be set for both the figure and figcaption, as:

```
<style>
figure {
  display: block;
  margin-top: 1em;
  margin-bottom: 1em;
  margin-left: 40px;
  margin-right: 40px;
}
</style>
```



Background Styles

- The use of images for backgrounds is supported by CSS using the following `background-image` style:

`background-image: url(url);`

where `url` specifies the name and location of the background image

- **Tiling** is the process of repeating an image both vertically and horizontally until the background is filled
- The type of tiling can be specified by applying the following `background-repeat` style:

`background-repeat: type;`

where `type` can be `repeat` (the default), `repeat-x`, `repeat-y`, `round`, or `space`

Repeating An Image

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
  background-repeat: repeat-y;
}
</style>
</head>
<body>

<h1>The background-repeat Property</h1>
<p>Here, the background image is repeated only vertically.</p>

</body>
</html>
```

The background-repeat Property

Here, the background image is repeated only vertically.



Repeating An Image (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
  background-repeat: repeat-x;
}
</style>
</head>
<body>

<h1>The background-repeat Property</h1>
<p>Here, the background image is repeated only horizontally.</p>

</body>
</html>
```

The background-repeat Property

Here, the background image is repeated only horizontally.

Repeating An Image (con't)

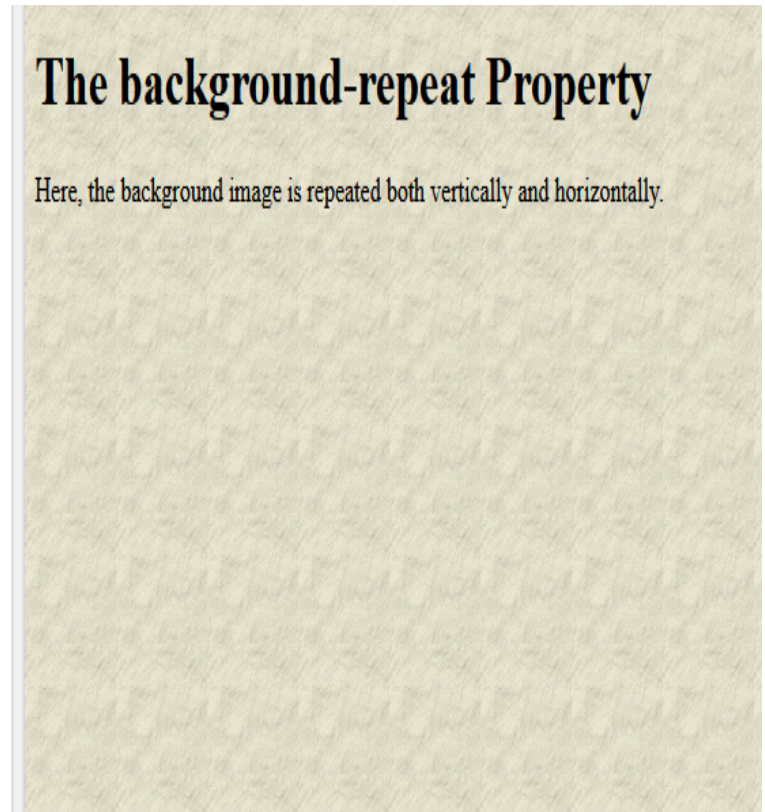
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
  background-repeat: repeat;
}
</style>
</head>
<body>

<h1>The background-repeat Property</h1>
<p>Here, the background image is repeated both vertically and horizontally.</p>

</body>
</html>
```

The background-repeat Property

Here, the background image is repeated both vertically and horizontally.



Repeating An Image (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("paper.gif");
  background-repeat: no-repeat;
}
</style>
</head>
<body>

<h1>The background-repeat Property</h1>
<p>Here, the background image is not repeated. The image will only be shown once.</p>

</body>
</html>
```

The background-repeat Property

Here, the background image is not repeated. The image will only be shown once.

Background Styles (con't)

- A background image is attached to its element so that it scrolls when the element content is scrolled
- The attachment can be changed using the following property:

`background-attachment: type;`

where *type* is `scroll` (the default), `fixed`, or `local`

- *type* in the background-attachment property:
 - `scroll` sets the background to scroll with the element content
 - `fixed` creates a background that stays in place even as the element content is scrolled
 - `local` allows the element background to scroll along with the content within the box

Background Image Position

- **Watermarks** are translucent graphics displayed behind a content
- They can be created using fixed backgrounds
- They are often used to indicate that a content material is copyrighted
- **By default, background images are placed in an element's top-left corner**
- The following property can be used to set the position of a background image:

`background-position: horizontal vertical;`

where `horizontal` and `vertical` provide the coordinates of an image within the element background

Background Image Position (con't)

- Keywords to position a background image are as follows:
 - left, center, and right are used to position the background horizontally
 - top, center, and bottom are used to position the background vertically
- The background-position property is only useful for non-tiled images

Background Image Clip

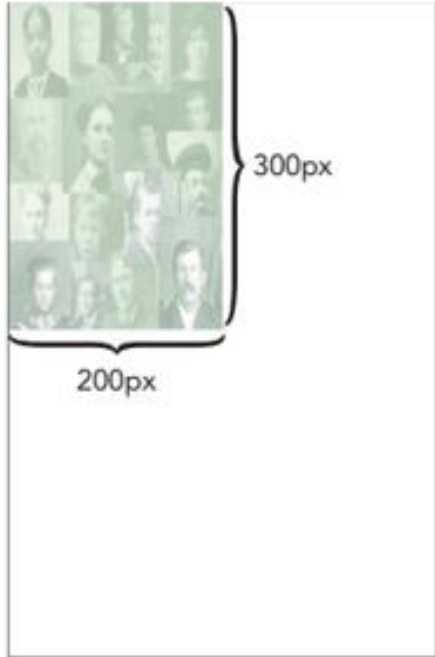
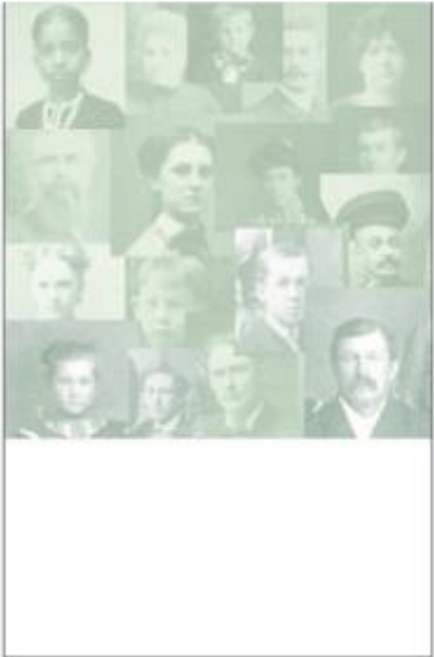
- An element's background extends only through the padding space excluding the border space
- This can be changed using the following property:
`background-clip: type;`
- `type` in the `background-clip` property:
 - `content-box` extends the background only through the element content
 - `padding-box` extends the background through the padding space
 - `border-box` extends the background through the border space

Background Image Size

- By default, the size of a background image equals the size stored in its image file
- This size can be changed by using the following property:

`background-size: width height;`
where *width* and *height* can be specified in pixels or by using the keywords `auto`, `cover`, and `contain`

Background Image Size (con't)

<code>background-size: 200px 300px;</code>	<code>background-size: cover;</code>	<code>background-size: contain;</code>
		
image is scaled at the specified dimensions	image is resized to fill the background, but part of the image is cropped	image is resized so that it is contained within the element, but part of the background is left uncovered

Background Styles (con't)

- Different background options can be organized using the following property:

```
background: color url(url) position/size repeat  
attachment origin clip;
```

where `color` is the background color, `url` is the source of the background image, and `position` is the image's position

`size` sets the image size

`repeat` sets the tiling of the image

`attachment` specifies whether the image scrolls with the content or is fixed

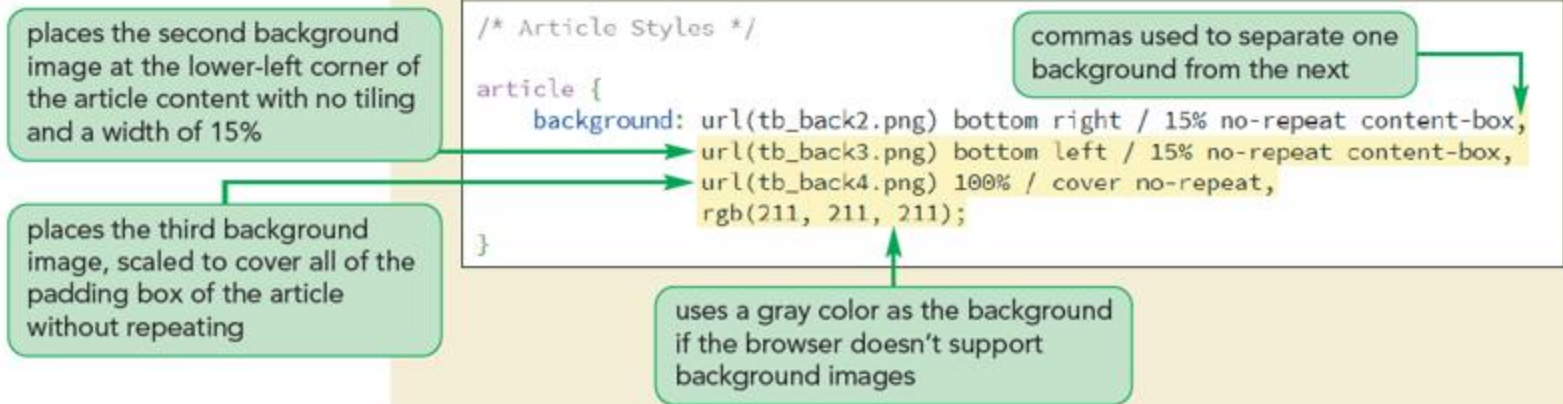
`origin` defines how positions are measured on the background

`clip` specifies the extent over which the background is spread

Background Styles (con't)

- **Multiple backgrounds** can be added to a single element by listing the backgrounds in the following comma-separated list:

background: background1, background2,
...;



Borders

- There are several style properties that can be used to format the border around each element using CSS
- To define the thickness of a specific border, the following property can be used:

```
border-side-width: width;
```

where `side` is either `top`, `right`, `bottom`, or `left` and `width` is the width of the border in one of the CSS units of measure

- The appearance of borders can be further defined by using the following style:

```
border-side-style: style;
```

where `side` indicates the border side and `style` specifies one of the nine border styles

Borders (con't)

adds a 1-pixel solid gray border to the left and right edges of the page body

```
/* Page Body Styles */  
  
body {  
  border-left: 1px solid rgb(51, 51, 51);  
  border-right: 1px solid rgb(51, 51, 51);  
}
```

adds a 4-pixel double medium green border to the aside element

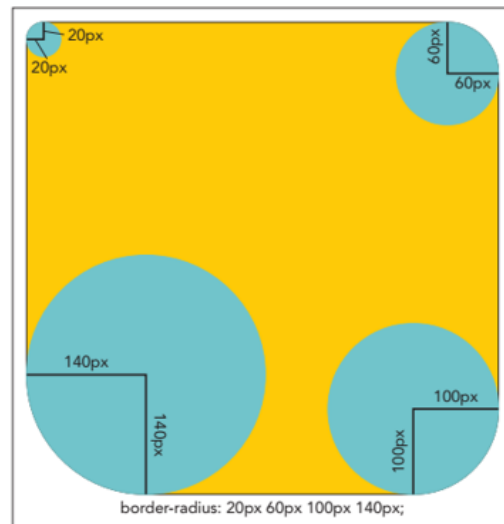
```
/* Aside Styles */  
  
aside {  
  border: 4px double rgb(45, 93, 62);  
}
```

Borders (con't)

- Any of the four corners of a border can be rounded off by applying the following property:

```
border-radius: top-left top-right bottom-right  
bottom-left;
```

where `top-left`, `top-right`, `bottom-right`, **and** `bottom-left` are the radii of the individual corners



Borders (con't)

sets the radius at each border corner to 30 pixels

```
aside {  
  border: 4px double rgb(45, 93, 62);  
  border-radius: 30px;  
}
```

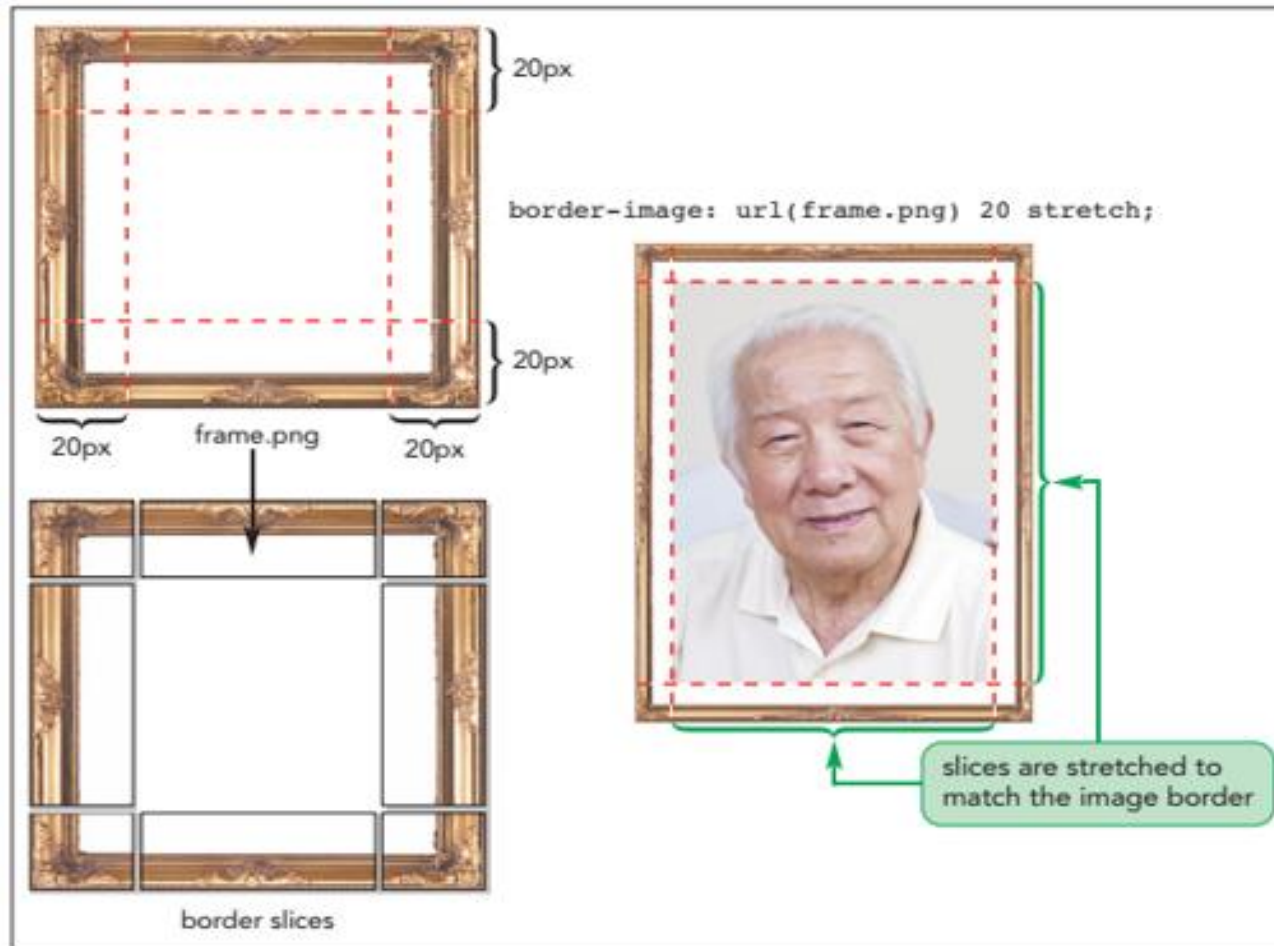
rounded corner



Borders (con't)

- A border image is a border based on a graphic image
- The graphic image is sliced into nine sections representing the four corners, the four sides, and the interior piece
- The content of the object appears in the interior piece and this piece is discarded
- The four corners become the border corners
- The four sides are either stretched or tiled to fill in the border's top, right, bottom, and left sides

Borders (con't)



Borders (con't)

- Border image uses the following property:

```
border-image: url(url) slice repeat fill;
```

where *url* is the source of the graphic image, *slice* is the width or height of the slices used to create the sides and corners, *repeat* indicates whether the side slices should be stretched or tiled to cover the border's four sides, and *fill* is an optional attribute that fills the image background with the graphic image file

Borders (con't)

- The *repeat* option supports:
 - `stretch`: slices are stretched to fill each side
 - `repeat`: slices are tiled to fill each side
 - `round`: when the slices are tiled to fill each side, if they do not fill the sides with an integer number of tiles, the slices are rescaled until they do
 - `space`: when the slices are tiled to fill each side, if they do not fill the sides with an integer number of tiles, extra space is distributed around the tiles

Shadow Styles

- To give the text on a page visual impact, use CSS to add a shadow using the following `text-shadow` property:

```
text-shadow: color offsetX offsetY  
blur;
```

where `color` is the shadow color

`offsetX` and `offsetY` are the distances of the shadow from the text in the horizontal and vertical directions

`blur` defines the amount by which the shadow spreads out, creating a blurred effect

Shadow Styles (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
  text-shadow: 2px 2px #FF0000;
}
</style>
</head>
<body>

<h1>The text-shadow Property</h1>

<p><b>Note:</b> Internet Explorer 9 and earlier do not support the text-shadow
property.</p>

</body>
</html>
```

The text-shadow Property

Note: Internet Explorer 9 and earlier do not support the text-shadow property.

Shadow Styles (con't)

```
article {  
  background: url(tb_back2.png) bottom right / 15% no-repeat content-box,  
             url(tb_back3.png) bottom left / 15% no-repeat content-box,  
             url(tb_back4.png) 100% / cover no-repeat,  
             rgb(211, 211, 211);  
}
```

```
article header h1 {  
  text-shadow: rgb(181, 211, 181) 2px 2px 1px,  
              rgba(21, 21, 21, 0.66) 5px 5px 25px;  
}
```

light green text
shadow with hard
edges

semi-transparent
gray shadow
with soft edges

shadow
color

horizontal
offset

vertical
offset

blur
size

Shadow Styles (con't)

- Any block element can be shadowed by using the `box-shadow` property

```
box-shadow: color offsetX  
           offsetY blur;
```

where `color`, `offsetX`, `offsetY`, and `blur` have the same meanings for box shadows as they do for text shadows

- Multiple shadows can be added by including them in a comma-separated list

Shadow Styles (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
#example1 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px;
}
#example2 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px #888888;
}
#example3 {
  border: 1px solid;
  padding: 10px;
  box-shadow: 5px 10px red;
}
</style>
</head>
<body>
<h1>The box-shadow Property</h1>
<p>The box-shadow property defines the shadow of an element.</p>
<h2>box-shadow: 5px 10px:</h2>
<div id="example1">
  <p>A div element with a shadow. The first value is the horizontal offset and the second value is the vertical offset. The shadow color will be inherited from the text color.</p>
</div>
<h2>box-shadow: 5px 10px #888888:</h2>
<div id="example2">
  <p>You can also define the color of the shadow. Here the shadow color is grey.</p>
</div>
<h2>box-shadow: 5px 10px red:</h2>
<div id="example3">
  <p>A red shadow.</p>
</div></body></html>
```

The box-shadow Property

The box-shadow property defines the shadow of an element:

box-shadow: 5px 10px:

A div element with a shadow. The first value is the horizontal offset and the second value is the vertical offset. The shadow color will be inherited from the text color.

box-shadow: 5px 10px #888888:

You can also define the color of the shadow. Here the shadow color is grey.

box-shadow: 5px 10px red:

A red shadow.

Shadow Styles (con't)

drop shadow on the left edge

drop shadow on the right edge

drop shadow on the page body's right edge

drop shadow on the page body's left edge

```
body {  
  border-left: 1px solid rgb(51, 51, 51);  
  border-right: 1px solid rgb(51, 51, 51);  
  box-shadow: rgb(51, 51, 51) 15px 0px 25px,  
             rgb(51, 51, 51) -15px 0px 25px;  
}
```

gray shadow color

```
article {  
  background: url(tb_back2.png) bottom right / 15% no-repeat content-box,  
             url(tb_back3.png) bottom left / 15% no-repeat content-box,  
             url(tb_back4.png) 100% / cover no-repeat,  
             rgb(211, 211, 211);  
  box-shadow: inset rgb(71, 71, 71) -10px -10px 25px,  
             inset rgb(71, 71, 71) 10px 10px 25px;  
}
```

places a medium-gray shadow in the lower-right interior corner

inset keyword places shadow inside the object

places a medium-gray shadow in the upper-left interior corner

Color Gradient

- Color gradient:
 - One color gradually blends into another color or fades away if transparent colors are used
 - It can be used to modify a background color
- CSS allows two types of gradients
 - Linear Gradients (goes down/up/left/right/diagonally)
 - Radial Gradients (defined by their center)
- A **linear gradient** is a color gradient in which the background color transitions from a starting color to an ending color along a straight line

Color Gradient (con't)

- Linear gradients are created using the linear-gradient function:

```
linear-gradient(color1, color2, ...)
```

where `color1`, `color2`, and so on are the colors that blend into one another starting from `color1`, through `color2`, and onto the last color listed

- The following gradient starts with a solid red, solid green appears halfway through the gradient, and finishes with solid blue:

```
background: linear-gradient(red, green, blue)
```

Color Gradient (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 200px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(red, yellow);
}
</style>
</head>
<body>

<h1>Linear Gradient - Top to Bottom</h1>
<p>This linear gradient starts red at the top, transitioning to yellow at the
bottom:</p>

<div id="grad1"></div>

</body>
</html>
```

Linear Gradient - Top to Bottom

This linear gradient starts red at the top, transitioning to yellow at the bottom:



Color Gradient (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 200px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(to right, red , yellow);
}
</style>
</head>
<body>

<h1>Linear Gradient - Left to Right</h1>
<p>This linear gradient starts red at the left, transitioning to yellow (to the
right):</p>

<div id="grad1"></div>

</body>
</html>
```

Linear Gradient - Left to Right

This linear gradient starts red at the left, transitioning to yellow (to the right):



Color Gradient (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 200px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(to bottom right, red, yellow);
}
</style>
</head>
<body>

<h1>Linear Gradient - Diagonal</h1>
<p>This linear gradient starts red at top left, transitioning to yellow (at bottom
right):</p>

<div id="grad1"></div>

</body>
</html>
```



Linear Gradient - Diagonal

This linear gradient starts red at top left, transitioning to yellow (at bottom right):



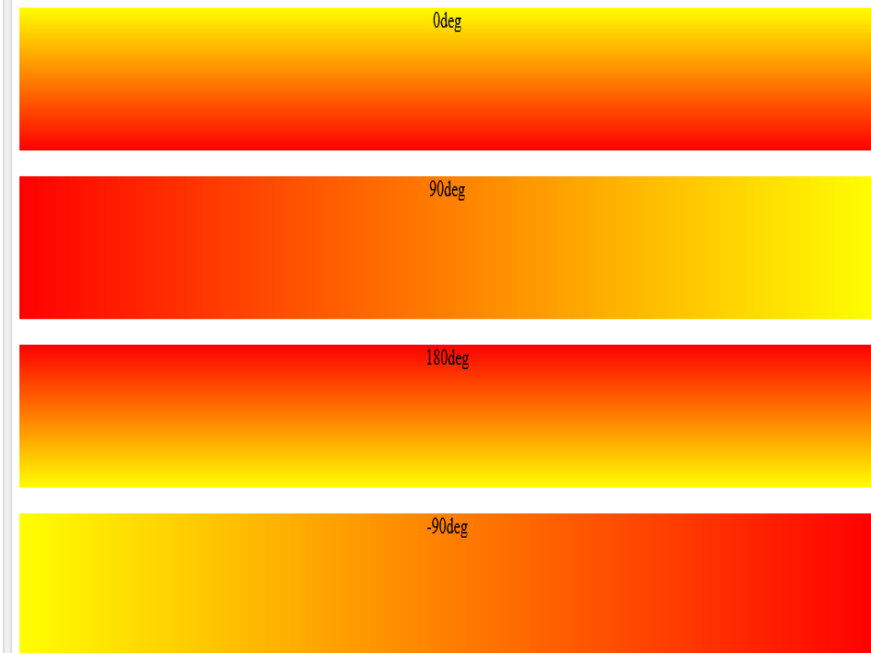
Color Gradient (con't)



Color Gradient (con't)

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 100px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(0deg, red, yellow);
}
#grad2 {
  height: 100px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(90deg, red, yellow);
}
#grad3 {
  height: 100px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(180deg, red, yellow);
}
#grad4 {
  height: 100px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: linear-gradient(-90deg, red, yellow);
}
</style>
</head>
<body>
<h1>Linear Gradients - Using Different Angles</h1>
<div id="grad1" style="text-align:center;">0deg</div><br>
<div id="grad2" style="text-align:center;">90deg</div><br>
<div id="grad3" style="text-align:center;">180deg</div><br>
<div id="grad4" style="text-align:center;">-90deg</div>
</body>
</html>
```

Linear Gradients - Using Different Angles



Color Gradient (con't)

- A **radial gradient** is a color gradient that starts from a central point and proceeds outward in a series of concentric circles or ellipses

- Radial gradients are created using:

```
radial-gradient(shape size at position,  
color-stop1, color-stop2, ...)
```

The `shape` value defines the shape of the gradient and is either `ellipse` (**default**) or `circle`

Color Gradient (con't)

- The `position` defines where the gradient radiates from and can be expressed in coordinates using pixels, percentages of the element's width and height, or with the keywords: `left`, `center`, `right`, `top`, and `bottom`
- The `default` is to place the gradient within the `center` of the background

Color Gradient (con't)

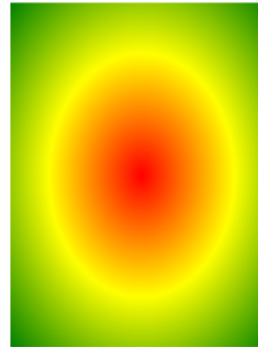
```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 150px;
  width: 200px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: radial-gradient(red, yellow, green);
}
</style>
</head>
<body>

<h1>Radial Gradient - Evenly Spaced Color Stops</h1>

<div id="grad1"></div>

</body>
</html>
```

Radial Gradient - Evenly Spaced Color Stops



Color Gradient (con't)

- The `color-stop1`, `color-stop2` ... values are the colors and their stopping positions within the gradient and have the same interpretation used for linear gradients except they mark stopping points as the gradient radiates outward
- The color stops are optional, just as they are in linear gradients

Color Gradient (con't)

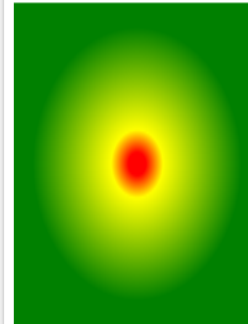
```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 150px;
  width: 200px;
  background-color: red; /* For browsers that do not support gradients */
  background-image: radial-gradient(red 5%, yellow 15%, green 60%);
}
</style>
</head>
<body>

<h1>Radial Gradient - Differently Spaced Color Stops</h1>

<div id="grad1"></div>

</body>
</html>
```

Radial Gradient - Differently Spaced Color Stops



Color Gradient (con't)

- The `size` value defines the extent of the gradient as it radiates outward and can be expressed with a CSS unit of measure, a percentage of the background's width and height, or with one of the following keywords:
 - `farthest-corner` (default): Gradient extends to the background corner farthest from the gradient's center
 - `farthest-side`: Gradient extends to background side farthest from the gradient's center
 - `closest-corner`: Gradient extends to the nearest background corner
 - `closest-side`: Gradient extends to the background side closest to the gradient's center

Opacity

- Semi-Transparent colors can be created using the following opacity property:

`opacity: value;`

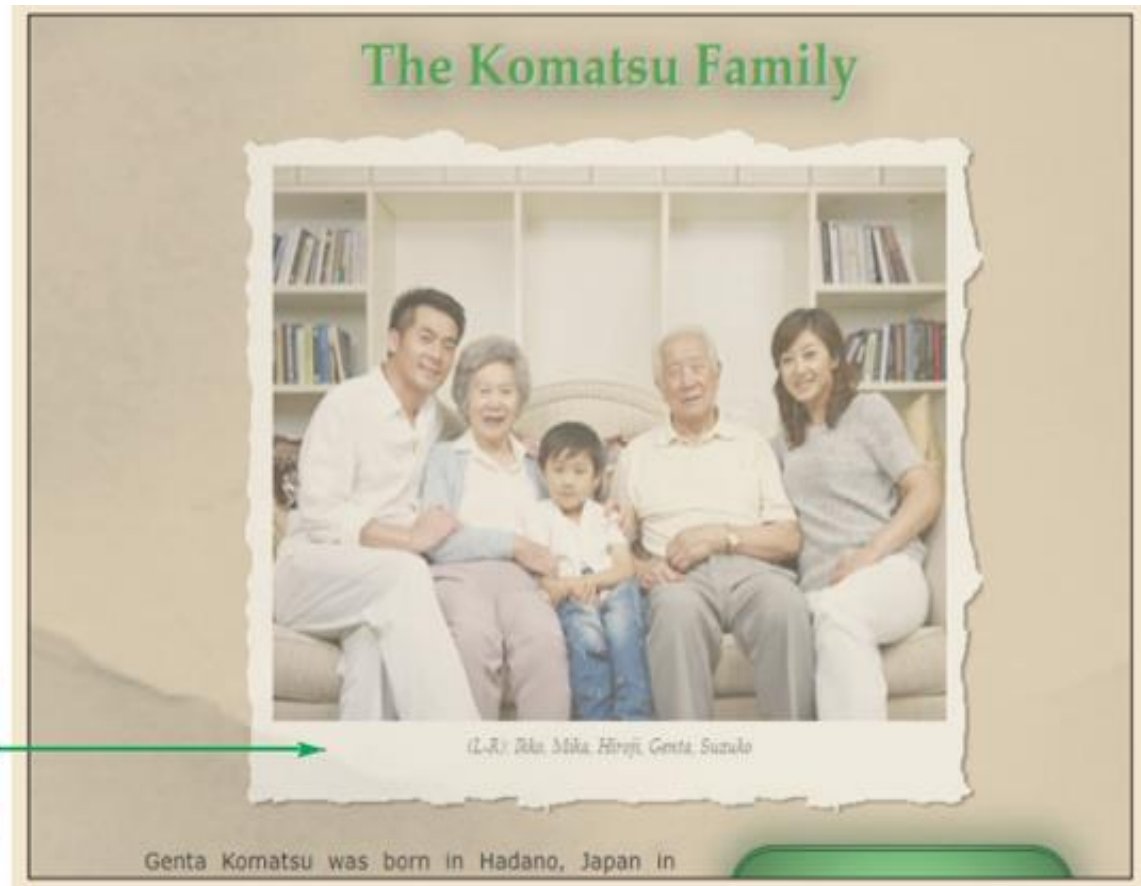
where `value` ranges from 0 (completely transparent) up to 1 (completely opaque)

sets the opacity of
the figure box to 55%

```
figure {  
  border-style: solid;  
  border-width: 25px;  
  border-image: url(tb_border.png) 50 repeat;  
  margin: 20px auto 0px;  
  opacity: 0.55;  
  width: 80%;  
}
```

Opacity (con't)

part of the
background
page texture
shows through
in the figure box



Transformations

- Rotation, rescaling, and translation in space can be used to transform the appearance of page objects

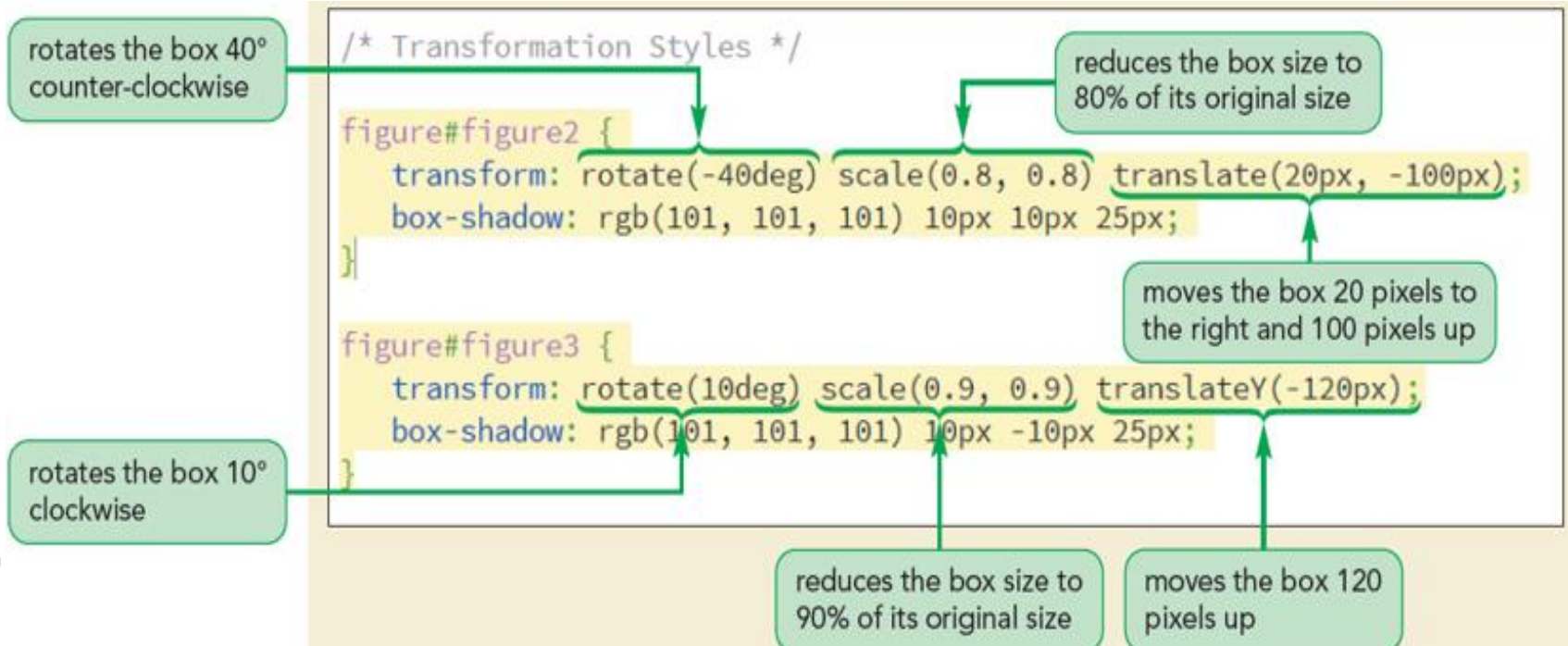
`transform: effect(params);`

where `effect` is the transformation function that will be applied to page objects and `params` specify the parameters required by the transformation function

Transformations (con't)

Function	Description
<code>translate(offX, offY)</code>	Moves the object <i>offX</i> pixels to the right and <i>offY</i> pixels down; negative values move the object to the left and up
<code>translateX(offX)</code>	Moves the object <i>offX</i> pixels to the right; negative values move the object to the left
<code>translateY(offY)</code>	Moves the object <i>offY</i> pixels down; negative values move the object up
<code>scale(x, y)</code>	Resizes the object by a factor of <i>x</i> horizontally and a factor of <i>y</i> vertically
<code>scaleX(x)</code>	Resizes the object by a factor of <i>x</i> horizontally
<code>scaleY(y)</code>	Resizes the object by a factor of <i>y</i> vertically
<code>skew(angleX, angleY)</code>	Skews the object by <i>angleX</i> degrees horizontally and <i>angleY</i> degrees vertically
<code>skewX(angleX)</code>	Skews the object by <i>angleX</i> degrees horizontally
<code>skewY(angleY)</code>	Skews the object by <i>angleY</i> degrees vertically
<code>rotate(angle)</code>	Rotates the object by <i>angle</i> degrees clockwise; negative values rotate the object counter-clockwise
<code>matrix(n, n, n, n, n, n)</code>	Applies a 2D transformation based on a matrix of six values

Transformations (con't)



Transformations (con't)

- A **3D transformation** is a change that involves three spatial axes:
 - an x-axis that runs horizontally across the page
 - a y-axis that runs vertically
 - a z-axis that comes straight out of the page toward and away from the viewer

Transformations (con't)

Function	Description
<code>translate3d(offX, offY, offZ)</code>	Shifts the object <i>offX</i> pixels horizontally, <i>offY</i> pixels vertically, and <i>offZ</i> pixels along the z-axis
<code>translateX(offX)</code> <code>translateY(offY)</code> <code>translateZ(offZ)</code>	Shifts the object <i>offX</i> , <i>offY</i> , or <i>offZ</i> pixels along the specified axis
<code>rotate3d(x, y, z, angle)</code>	Rotates the object around the three-dimensional vector (<i>x</i> , <i>y</i> , <i>z</i>) at a direction of <i>angle</i>
<code>rotateX(angle)</code> <code>rotateY(angle)</code> <code>rotateZ(angle)</code>	Rotates the object around the specified axis at a direction of <i>angle</i>
<code>scale3d(x, y, z)</code>	Resizes the object by a factor of <i>x</i> horizontally, a factor of <i>y</i> vertically, and a factor of <i>z</i> along the z-axis
<code>scaleX(x)</code> <code>scaleY(y)</code> <code>scaleZ(z)</code>	Resizes the object by a factor of <i>x</i> , <i>y</i> , or <i>z</i> along the specified axis
<code>perspective(p)</code>	Sets the size of the perspective effect to <i>p</i>
<code>matrix3d(n, n, ..., n)</code>	Applies a 3D transformation based on a matrix of 16 values

Perspective

- Perspective is a measure of how rapidly objects appear to recede from the viewer in a 3D space
- Perspective can be thought in terms of a pair of railroad tracks that appear to converge at a point, known as the **vanishing point**
- The perspective of a 3D space can be defined using

`perspective: value;`

where `value` is a positive value that measures the strength of the perspective effect

- Lower value results in more extreme distortion ⁵¹

Perspective (con't)

sets the perspective of the article space to 600 pixels

rotates the box 30° around the x-axis and shifts it forward 50 pixels along the z-axis

rotates the box 30° around the z-axis and 60° around the y-axis

rotates the box 70° counter-clockwise around the y-axis and shifts it backward 20 pixels along the z-axis

```
/* Transformation Styles */
```

```
article {  
  perspective: 600px;  
}
```

adds a box shadow on the box's bottom border

```
figure#figure1 {  
  transform: rotateX(30deg) translateZ(50px);  
  box-shadow: rgb(51, 51, 51) 0px 10px 25px;  
}
```

```
figure#figure2 {  
  transform: rotate(-40deg) scale(0.8, 0.8)  
    translate(20px, -100px)  
    rotateZ(30deg) rotateY(60deg);  
  box-shadow: rgb(101, 101, 101) 10px 10px 25px;  
}
```

```
figure#figure3 {  
  transform: rotate(10deg) scale(0.9, 0.9)  
    translateY(-120px)  
    rotateY(-70deg) translateZ(-20px);  
  box-shadow: rgb(101, 101, 101) 10px -10px 25px;  
}
```

Filters

- Filters adjust how a browser renders an image, a background, or a border
- Filters modify an object's color, brightness, contrast, or general appearance
- Filters can be applied using the following property:

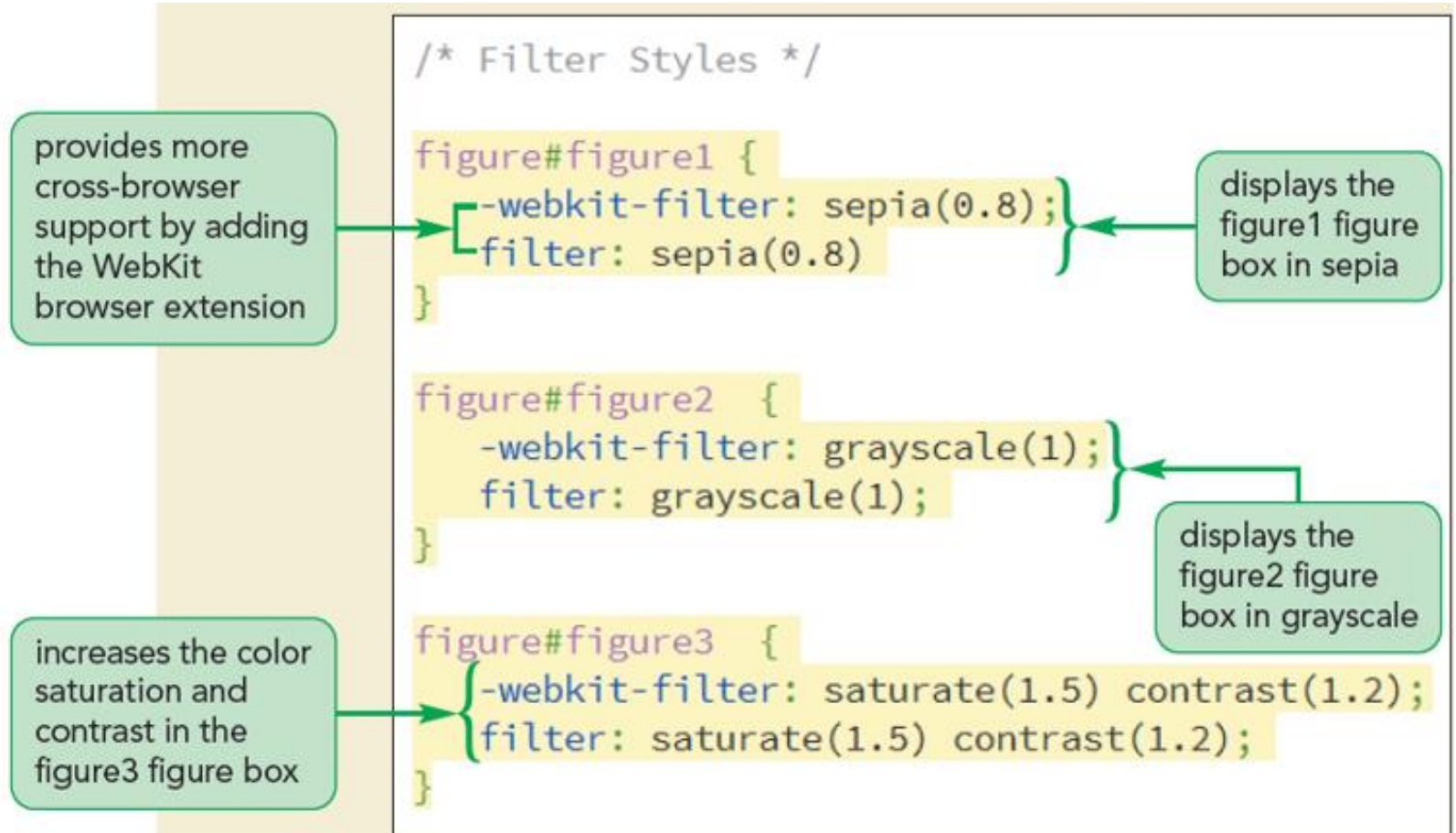
```
filter: effect(params);
```

where `effect` is a filter function and `params` specify the parameters of filter function

Filters (con't)

Function	Description
<code>blur(<i>length</i>)</code>	Applies a blur to the image where <i>length</i> defines the size of blur in pixels
<code>brightness(<i>value</i>)</code>	Adjusts the brightness where values from 0 to 1 decrease the brightness and values greater than 1 increase the brightness
<code>contrast(<i>value</i>)</code>	Adjusts the contrast where values from 0 to 1 decrease the contrast and values greater than 1 increase the contrast
<code>drop-shadow(<i>offsetX</i> <i>offsetY</i> <i>blur</i> <i>color</i>)</code>	Adds a drop shadow to the image where <i>offsetX</i> and <i>offsetY</i> are horizontal and vertical distances of the shadow, <i>blur</i> is the shadow blurring, and <i>color</i> is the shadow color
<code>grayscale(<i>value</i>)</code>	Displays the image in grayscale from 0, leaving the image unchanged, up to 1, displaying the image in complete grayscale
<code>hue-rotate(<i>angle</i>)</code>	Adjusts the hue by <i>angle</i> in the color wheel where 0deg leaves the hue unchanged, 180deg displays the complimentary colors and 360deg again leaves the hue unchanged
<code>invert(<i>value</i>)</code>	Inverts the color from 0 (leaving the image unchanged), up to 1 (completely inverting the colors)
<code>opacity(<i>value</i>)</code>	Applies transparency to the image from 0 (making the image transparent), up to 1 (leaving the image opaque)
<code>saturate(<i>value</i>)</code>	Adjusts the color saturation where values from 0 to 1 decrease the saturation and values greater than 1 increase the saturation
<code>sepia(<i>value</i>)</code>	Displays the color in a sepia tone from 0 (leaving the image unchanged), up to 1 (image completely in sepia)
<code>url(<i>url</i>)</code>	Loads an SVG filter file from <i>url</i>

Filters (con't)



References

- Graphic Design For Everyone: Understand the Building Blocks so You can Do It Yourself by Cath Caldwell
- Web Design. The Evolution of the Digital World 1990–Today (multilingual Edition) by Rob Ford and Julius Wiedemann
- White Space Is Not Your Enemy: A Beginner's Guide to Communicating Visually Through Graphic, Web & Multimedia Design by Kim Golombisky and Rebecca Hagen

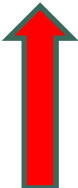
Homework

- Textbook Chapter 4
- Textbook Appendix C
- Apply some background style to you home page
 - See following slides →

Background's in John Doe's Page

[htmlab13.htm]

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>John Doe's Home Page</title>
    <style>
      /* structural styles */
      article, aside, footer, header, main, nav, section {
        display: block;
      }
      /* layout styles */
      header {margin-left:auto; margin-right:auto}
      nav {float:left; width:15%; border:3px solid gray; background-color:tan; padding:3px; margin:3px;}
      aside {float:right; width:15%; border:3px solid gray; background-color:tan; padding:3px; margin:3px;}
      section {border:3px; padding:6px; margin:3px;}
      footer {clear:both;}
      /* grid style */
      div#mygrid {display:grid;}
      div#address-cell {background: rgb(211,211,211); outline: 3px solid gray;}
      div#hobby-cell {background: rgb(211,211,211); outline: 3px solid gray;}
      div#girl-cell {background: rgb(211,211,211); outline: 3px solid gray;}
      div#pet-cell {background: rgb(211,211,211); outline: 3px solid gray;}
    </style>
  </head>
```



John Doe



సుసావగత (Hello in Japanese)

This is a picture of me.

I am so happy to be in this class.

I always do my reading and homework on time, and I never miss a class.

Links:

- [address](#)
- [hobbies](#)
- [girlfriend](#)
- [pets](#)

My address

*11 Peach Street
Memphis, TN 38108*

My Hobbies

My hobby is going to the gym.

I take my text books with me to read while I use the treadmill.

I tried to study while in the spa, but I dropped my homework in the water.

I wish they would put internet terminals on the treadmills and stationery bikes.

Picture of Me at Gym



My Girlfriend

This is my girlfriend !

These are the things we like to do:

- Ride in my 1983 Ford Pinto
- Help pick up litter on our campus
- Eat in the school snack bar
- Sign up for 8am classes together
- Discuss student affairs with our school administration
- Study our math class notes together « What Fun »
- Surf the Net !

Picture of My Girlfriend



My Pets

Pet Types and Names

Name	Type
Fred	Pigeon
Alice	Skunk
Mike	Alligator

My Favorite Classes

- MIS 470
- MIS 471
- MIS 351
- MIS231