Computer Science & Information Technology
School of Fundamental Sciences

Graphic Design Using CSS3

Course textbook: Tutorial 4, p.257

Topic Scenario: *Tree and Book*

Kevin Whitmore is the founder of *Tree and Book,* a social networking website for people interested in documenting their family histories, creating online photo albums, and positive stories and information about members of their extended families. He has come to you for help in upgrading the site's design. Kevin wants to take advantage of some of the CSS styles that can be used to add interesting visual effects to his site in order to give his website more impact and visual interest.

Sample Solution: *Tree and Book Website*

MASSEY UNIVERSITY

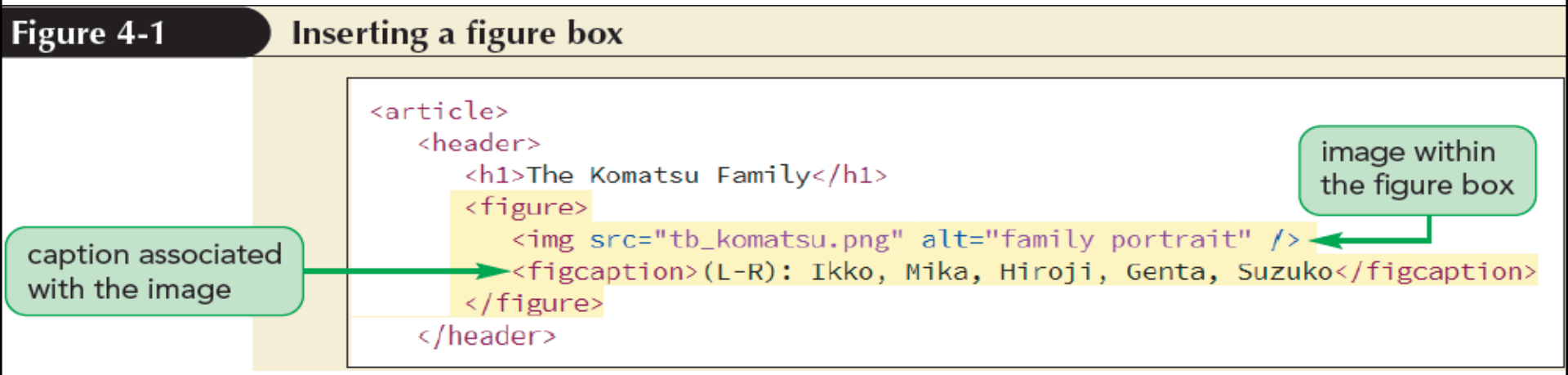
Learning Objectives

1. Create a figure box
2. Add a background image
3. Add a border to an element
4. Create rounded borders
5. Create a graphic border
6. Create a text shadow
7. Create a box shadow
8. Create linear and radial gradients
9. Set the opacity of an element
10. Apply a 2D and 3D transformation
11. Apply a CSS filter
12. Create an image map

Creating Figure Boxes

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

Figure 4-1 Inserting a figure box



Background Styles

```
background-image:
url(url);
```

Tiling a Background Image

```
background-repeat: type;
```

Figure 4-6 Tiled background image in the browser window

Attaching the Background Image

```
background-attachment:
type;
```

Setting the Background Image Position

```
background-position:
horizontal vertical;
```

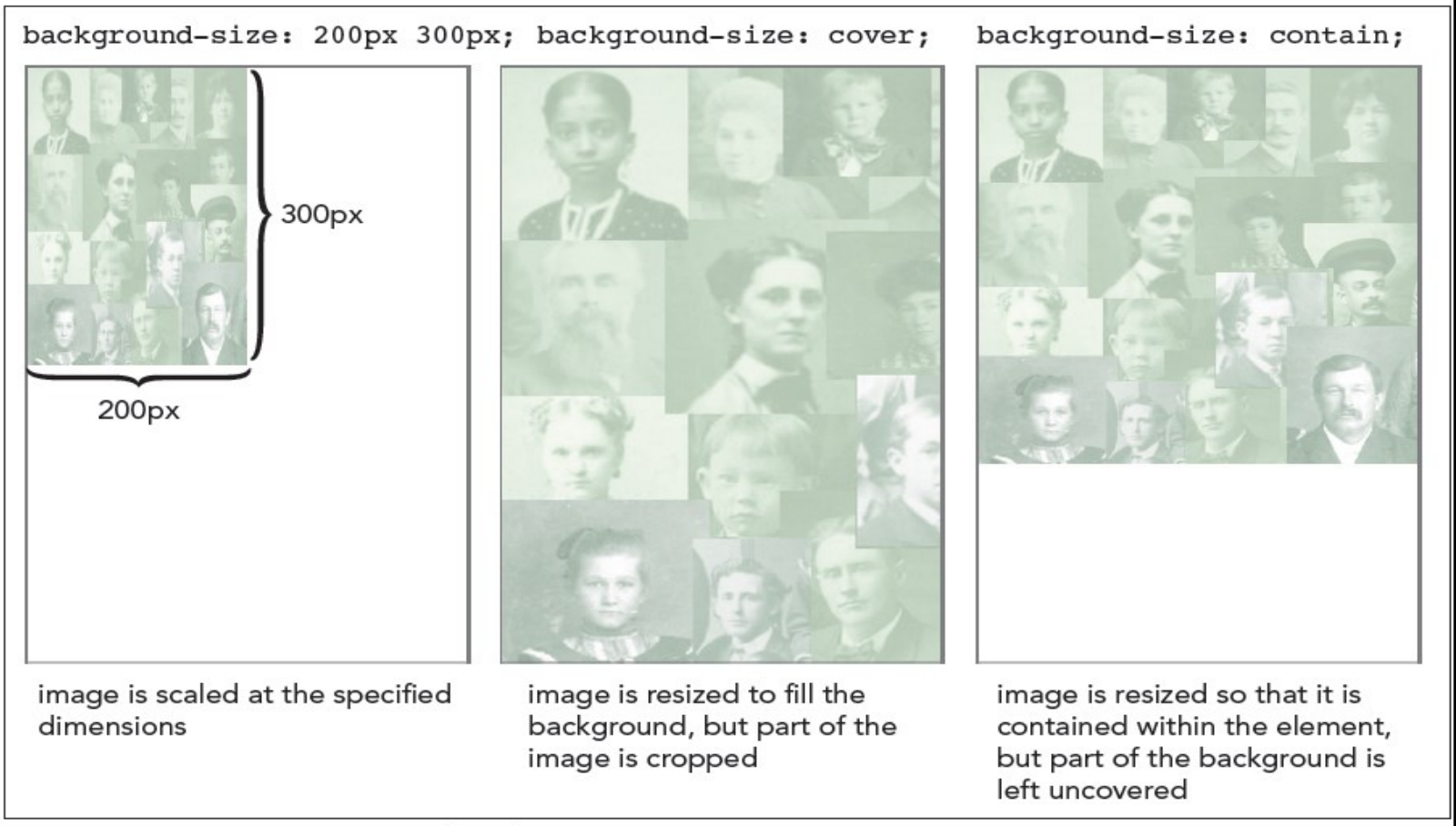
Defining the Extent of the Background

```
background-clip: type;
```

Sizing and Clipping an Image

```
background-size: width height;
```

Figure 4-7 Examples of background-size types



The `background` Property

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

Adding Multiple Backgrounds

`background: background1, background2, …;`

Figure 4-10 Adding multiple background images

places the second background image at the lower-left corner of the article content with no tiling and a width of 15%

places the third background image, scaled to cover all of the padding box of the article without repeating

```
/* Article Styles */

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);
}
```

commas used to separate one background from the next

uses a gray color as the background if the browser doesn't support background images

Setting Border Width and Color

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

Setting the Border Design

`border-side-style: style;`

Figure 4-13 Adding borders to the page body and aside element

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

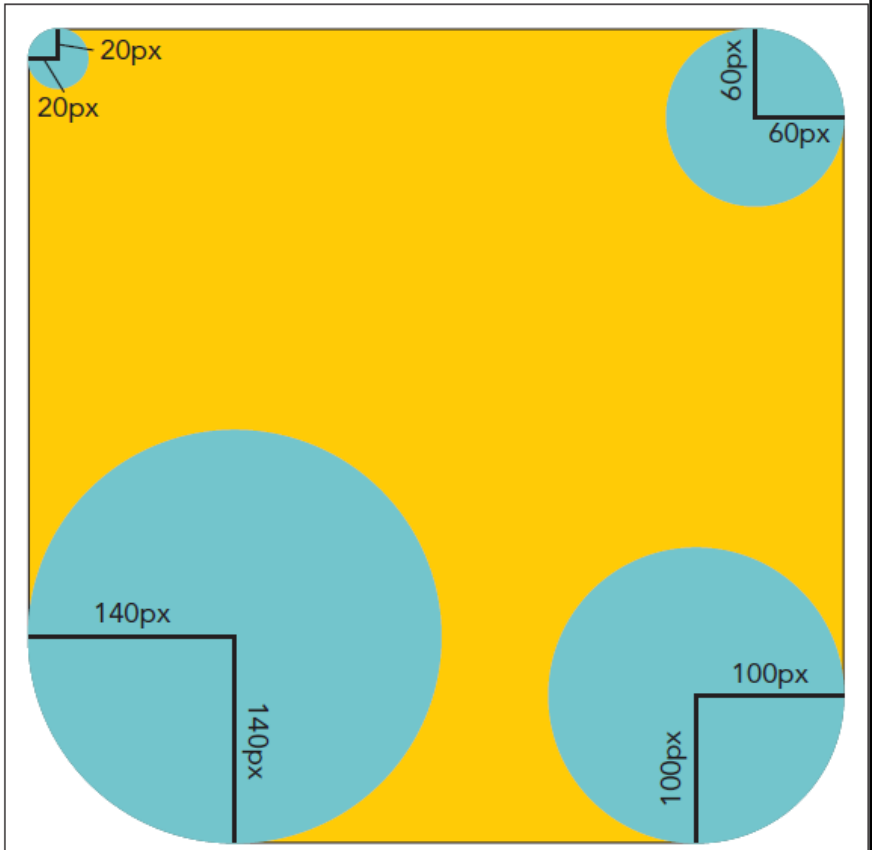
adds a 4-pixel double medium green border to the aside element

```
/* Page Body Styles */

body {
   border-left: 1px solid rgb(51, 51, 51);
   border-right: 1px solid rgb(51, 51, 51);
}


/* Aside Styles */

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

Creating Rounded Corners

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

Figure 4-15 Setting rounded corners based on corner radii



border-radius: 20px 60px 100px 140px;

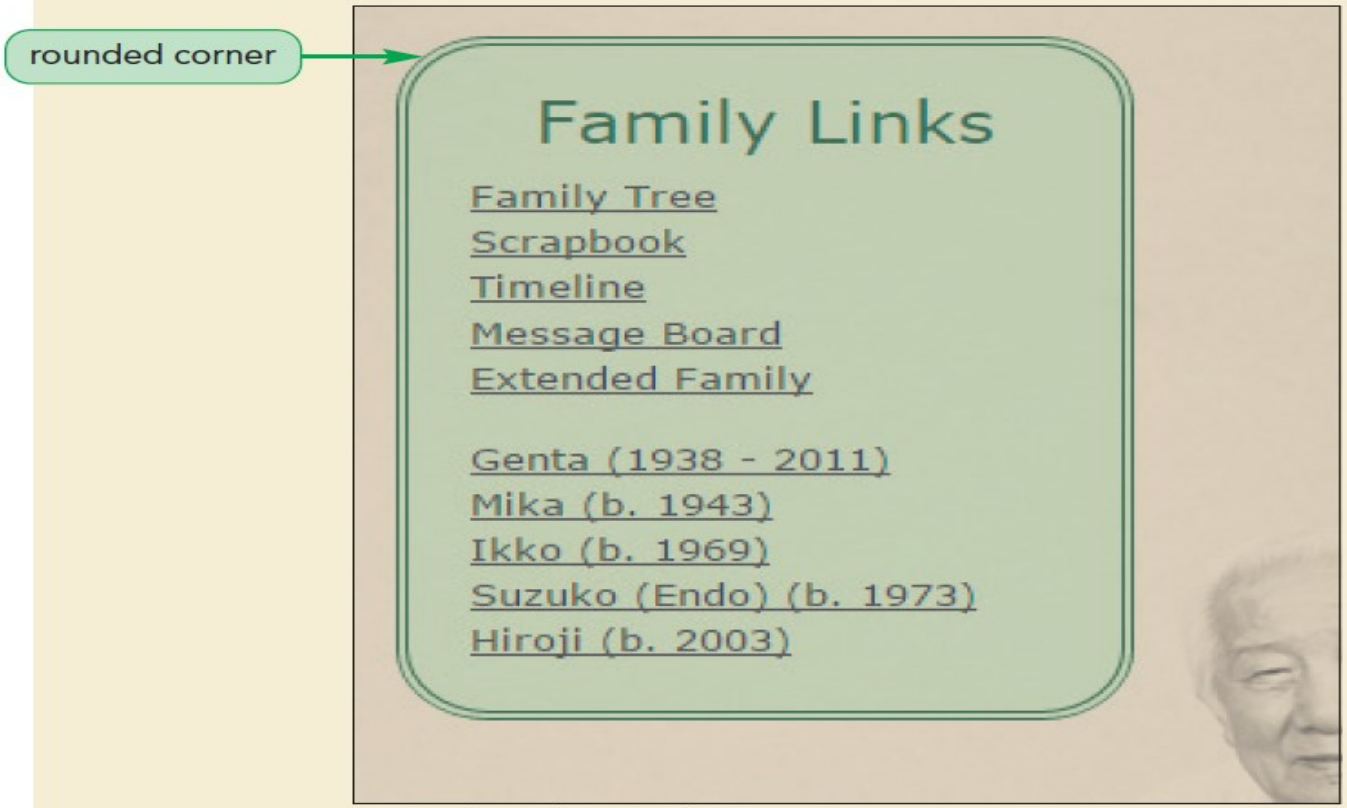
Creating Rounded Corners *(continued)*

Figure 4-17 Adding rounded corners to the aside element border

sets the radius at each border corner to 30 pixels

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

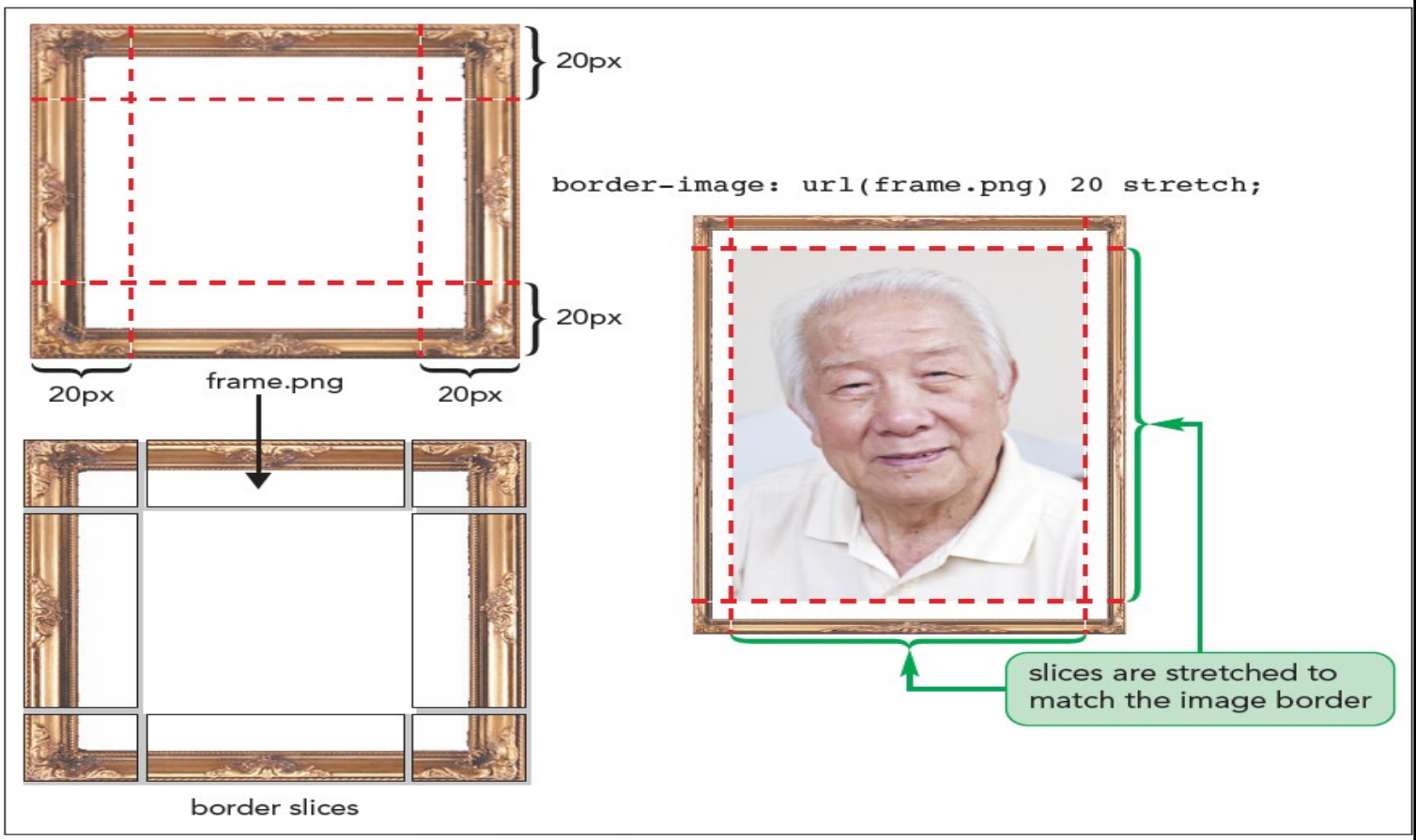
Figure 4-18 Aside element border with rounded corners



Applying a Border Image

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

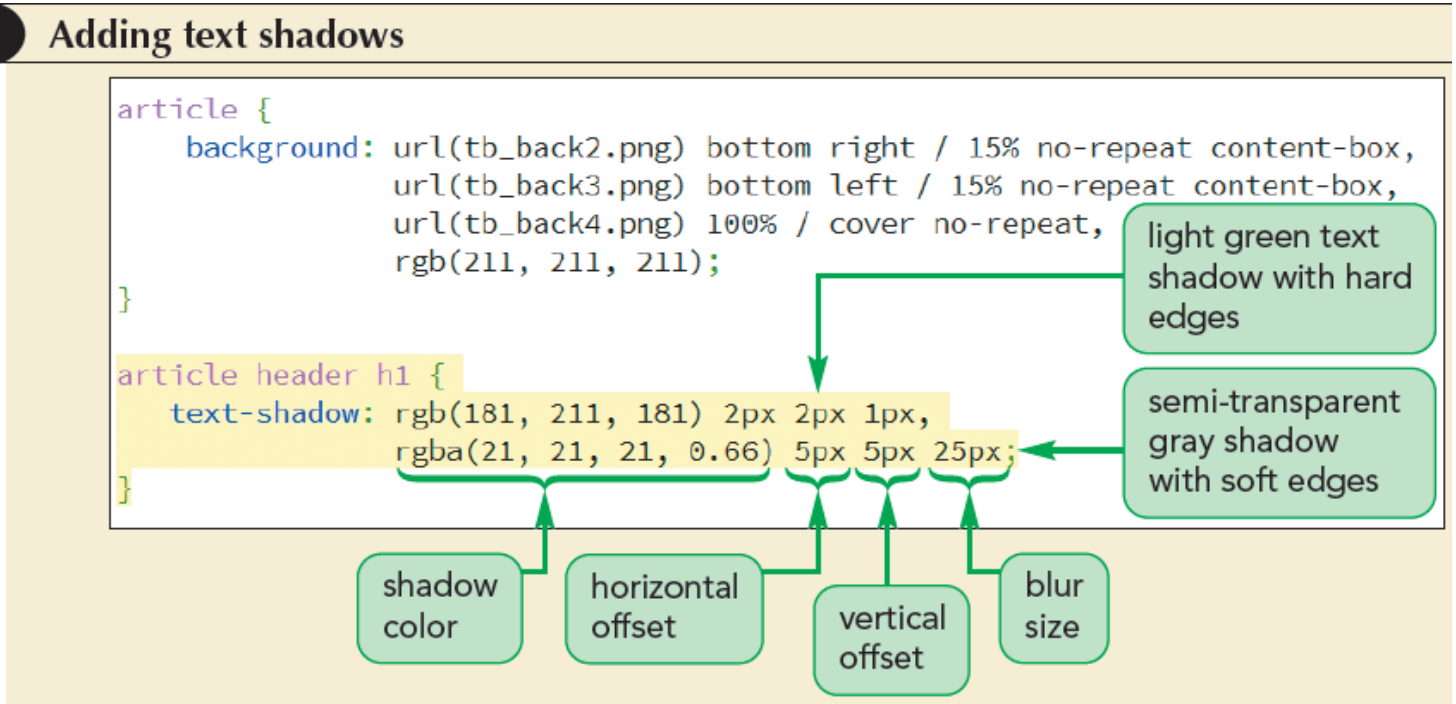
Figure 4-19 Slicing a graphic image to create a border



Creating a Text Shadow

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

Figure 4-23 Adding text shadows



Creating a Box Shadow

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

Figure 4-25 Adding box shadows

drop shadow on the pa

```
body {
```

Figure 4-26 Page body with drop shadows

Creating a Box Shadow *(continued)*

Figure 4-28 Adding an inset shadow

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

inset keyword places shadow inside the object

```
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;
}
```

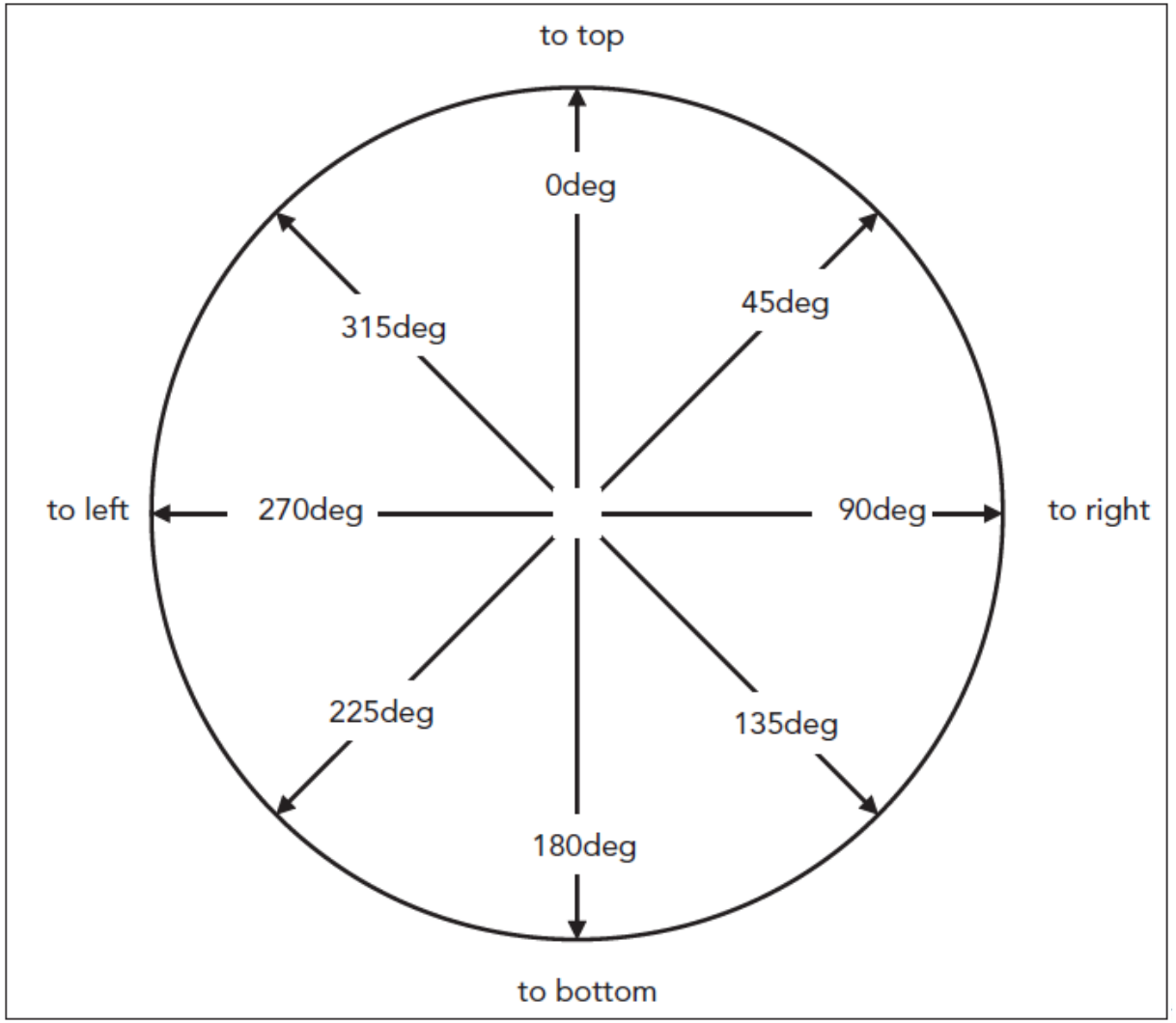
places a medium-gray shadow in the upper-left interior corner

Applying a Color Gradient

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

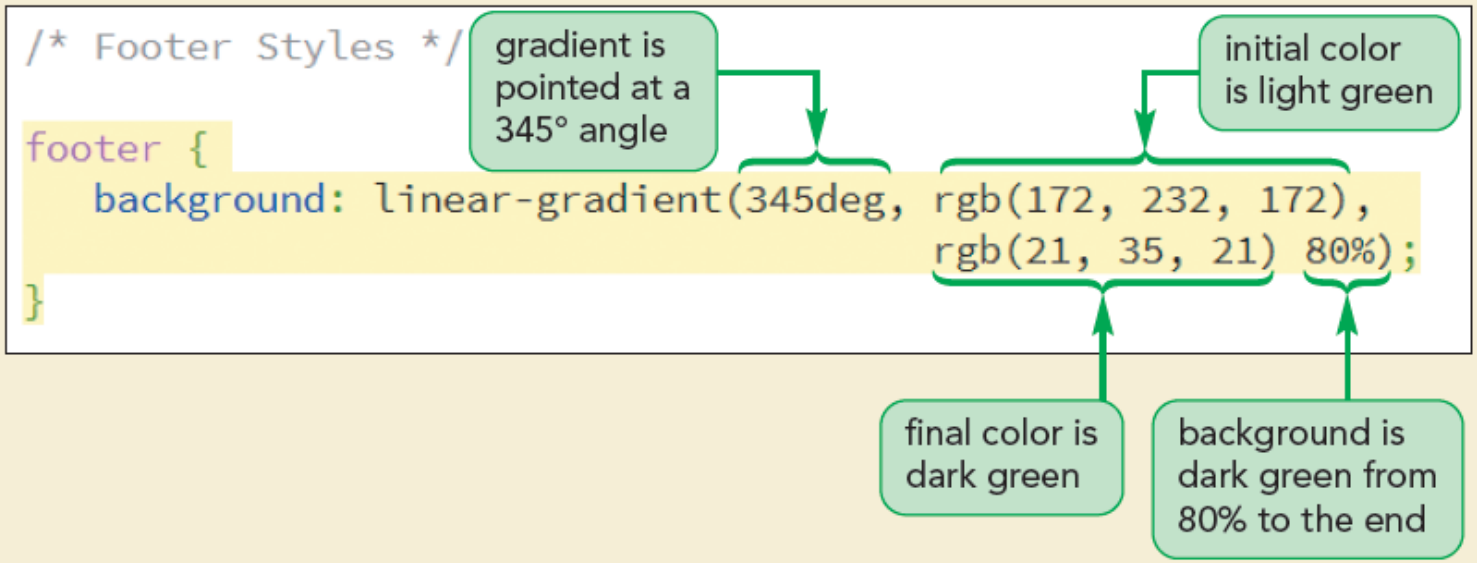
Gradients and Color Stops

Figure 4-33 Linear gradient directions



Gradients and Color Stops *(continued)*

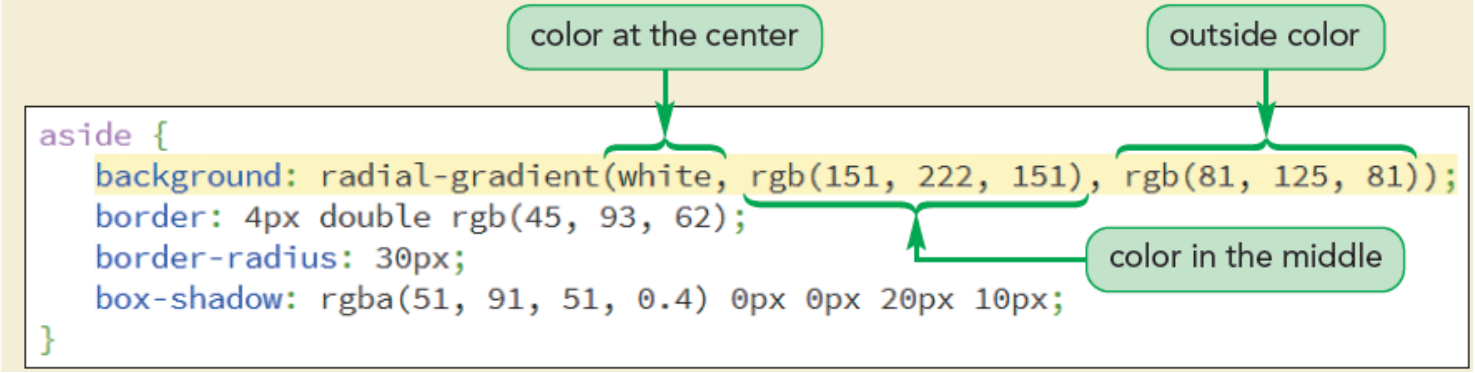
Figure 4-36 Applying a linear gradient



Creating a Radial Gradient

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

Figure 4-40 Applying a radial gradient



Repeating a Gradient

`repeating-linear-gradient(`*`params`*`)`

`repeating-radial-gradient(`*`params`*`)`

Creating Semi-Transparent Objects

Figure 4-43 Creating a semi-transparent object

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

sets the opacity of the figure box to 55%

Figure 4-44 Changing the opacity of the figure box

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

Transforming Page Objects

```
transform:
effect(params);
```

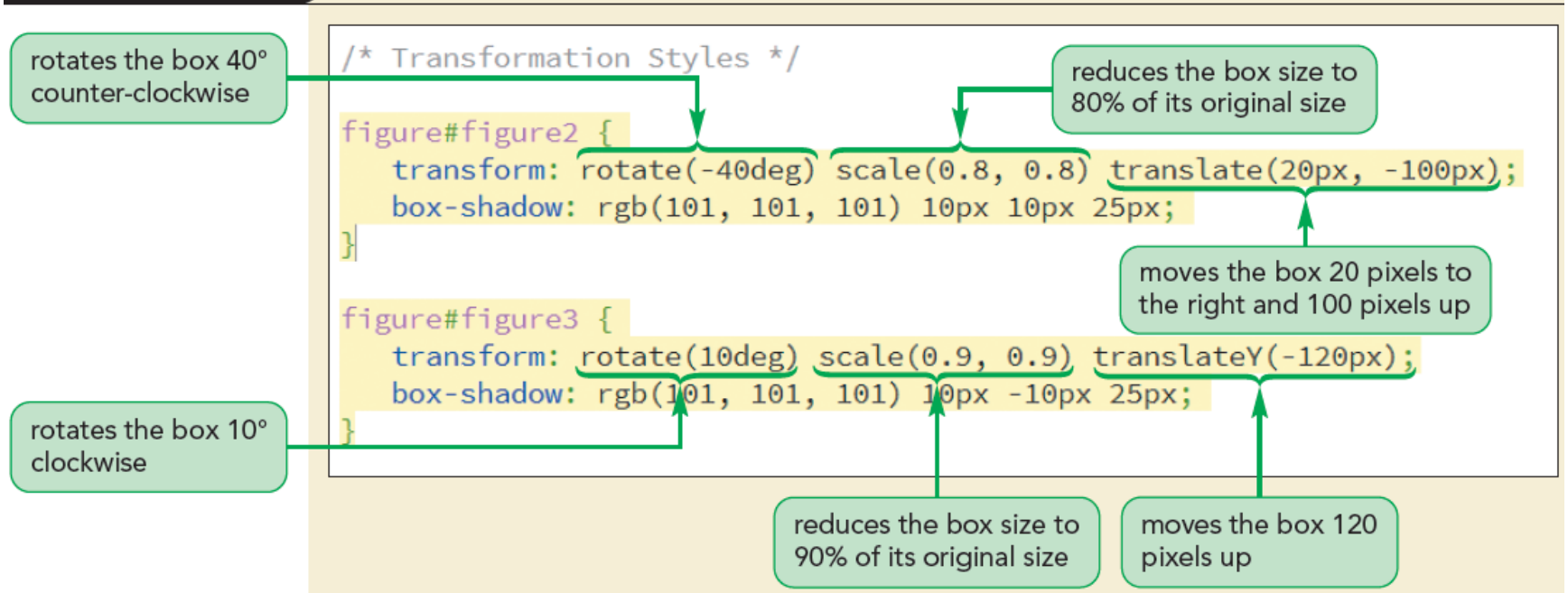
Transforming Page Objects *(continued 1)*

Figure 4-45 CSS3 2D transformation functions

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

Transforming Page Objects *(continued 2)*

Figure 4-48 Transforming the figure boxes



Transformations in Three Dimensions

Figure 4-51 CSS3 3D transformation functions

Function	Description
`translate3d(offX, offY, offZ)`	Shifts the object *offX* pixels horizontally, *offY* pixels vertically, and *offZ* pixels along the *z*-axis

`translateX(offX)`
`translateY(offY)`
`translateZ(offZ)`
	Shifts the object *offX*, *offY*, or *offZ* pixels along the specified axis

`rotate3d(x, y, z, angle)`	Rotates the object around the three-dimensional vector (*x*, *y*, *z*) at a direction of *angle*

`rotateX(angle)`
`rotateY(angle)`
`rotateZ(angle)`
	Rotates the object around the specified axis at a direction of *angle*

`scale3d(x, y, z)`	Resizes the object by a factor of *x* horizontally, a factor of *y* vertically, and a factor of *z* along the *z*-axis

`scaleX(x)`
`scaleY(y)`
`scaleZ(z)`
	Resizes the object by a factor of *x*, *y*, or *z* along the specified axis

`perspective(p)`	Sets the size of the perspective effect to *p*
`matrix3d(n, n, …, n)`	Applies a 3D transformation based on a matrix of 16 values

Understanding Perspective

`perspective: value;`

Figure 4-53 Applying 3D transformations

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

adds a box shadow on the box's bottom border

```
/* Transformation Styles */

article {
   perspective: 600px;
}

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;
}
```

Exploring CSS filters

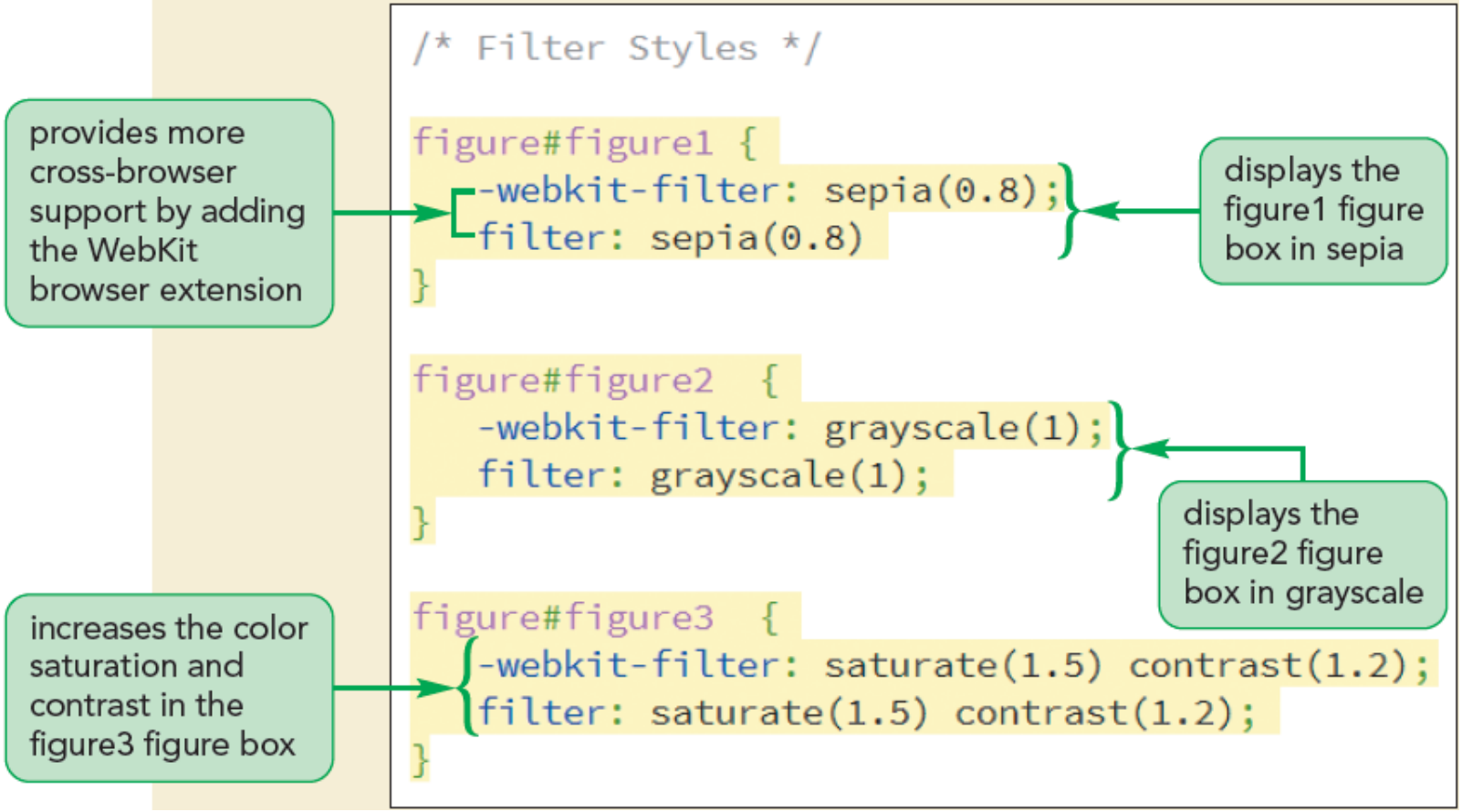
```
filter: effect(params);
```

Figure 4-55 CSS3 filter functions

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

Exploring CSS filters *(continued)*

Figure 4-57 Applying the filter property



Defining a Client-Side Image Map

```
<map name="text">
     hotspots
</map>
```

Defining hotspots:

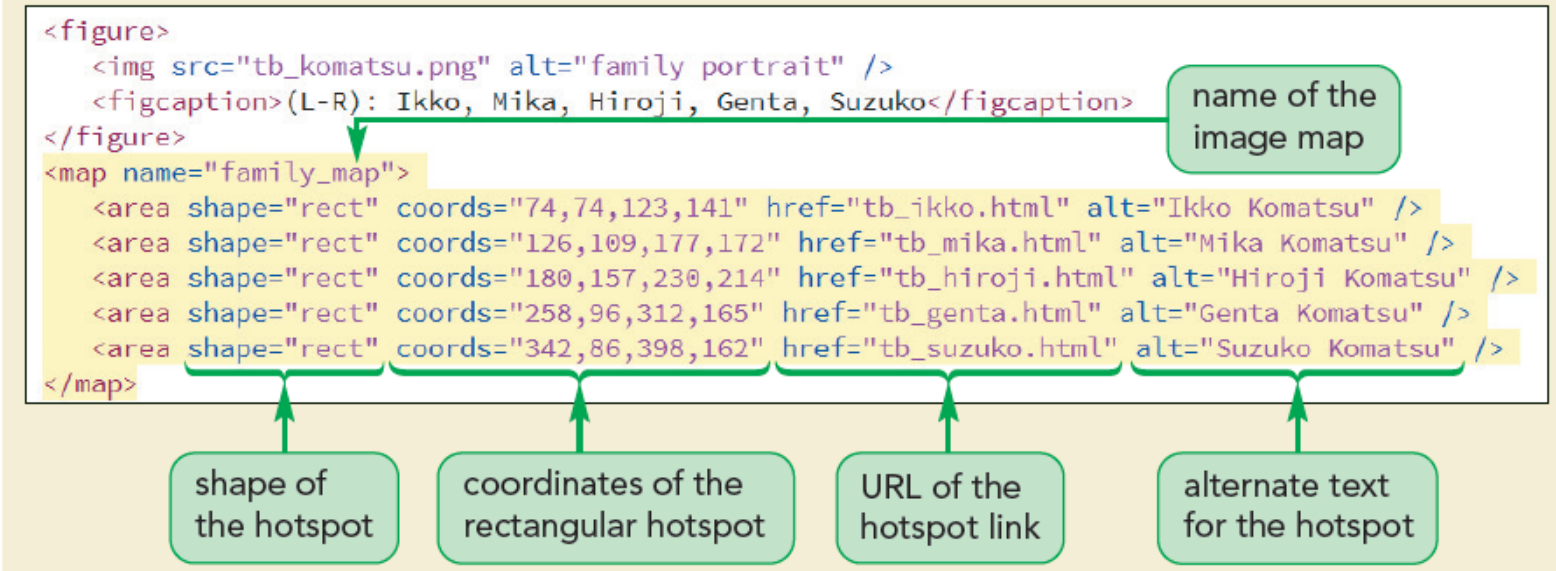
```
<area shape="shape" coords="coordinates"
      href="url" alt="text" />
```

Example circular hotspots:

```
shape="circle" coords="x, y, radius"
```

Defining a Client-Side Image Map *(continued)*

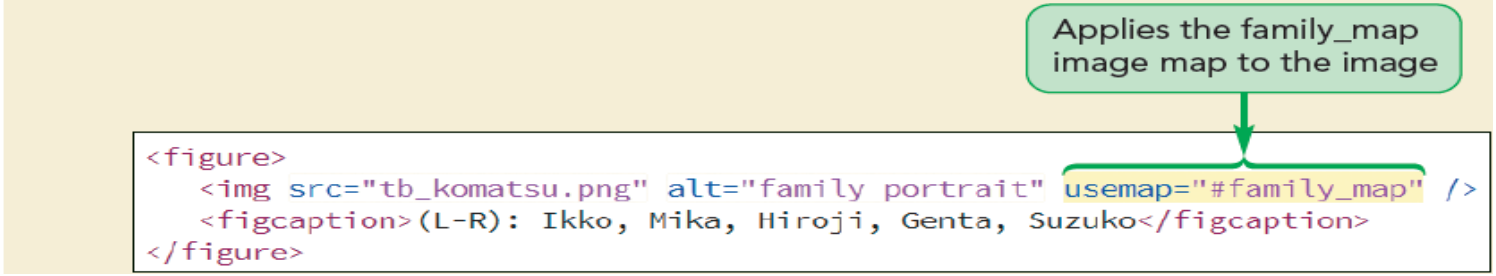
Figure 4-62 Inserting an image map



Applying an Image Map

```
<img src="url" alt="text"
usemap="#map" />
```

Figure 4-63 Applying an image map



Topic Summary

You will have mastered the material in this topic when you can:

- Session 4.1
  - ✓ Create a figure box
  - ✓ Add a background image
  - ✓ Add a border to an element
  - ✓ Create rounded borders
  - ✓ Create a graphic border
- Session 4.2
  - ✓ Create a text shadow
  - ✓ Create a box shadow
  - ✓ Create linear and radial gradients
  - ✓ Set the opacity of an element
- Session 4.3
  - ✓ Apply a 2D and 3D transformation
  - ✓ Apply a CSS filter
  - ✓ Create an image map

End of Lecture Slides: *Any Questions?*