Session: 07

Displaying Graphics and CSS3 Animation

Objectives

- Explain graphic formatting in Web pages
- Explain graphic insertion, sizing, and padding
- Explain CSS3 Animation
- Describe the use of CSS3 on mobile devices

Introduction

- After release of HTML5 and CSS3 in the market, Web designers have been developing graphics based Web pages.
- CSS3 has allowed designers to style their Web pages graphically with ease.
- Currently, HTML5 applications provide amazing experiences with the use of new CSS3 animations.
- Introduction of mobile applications has allowed users to expand their Web usage to mobile devices.
- CSS3 has introduced new features specifically for mobile devices.

Graphic Format 1-4

- There are many graphic formats available; the most commonly used are Joint Photographic Experts Group (JPEG), Graphics Interchange Format (GIF), and Portable Network Graphics (PNG).
- The difference between each graphic format depends on the following characteristics:

➢ Color Depth

- It is defined by the number of distinct colors that are represented by a hardware or software.
- Color depth is defined by the number of Bits Per Pixel (bpp) and it is also called as bit depth.
- Higher color depth indicates higher range of colors used.

➢ Compression/file size

- Compression stores the original images in a reduced number of bytes using an algorithm.
- This image can be expanded back to the original size using a decompression algorithm.

Graphic Format 2-4

- Two types of image file compression algorithms used are as follows:

Lossless compression

- File size is reduced, but preserves a copy of the original uncompressed image.
- Avoids accumulating stages of re-compression when editing images.

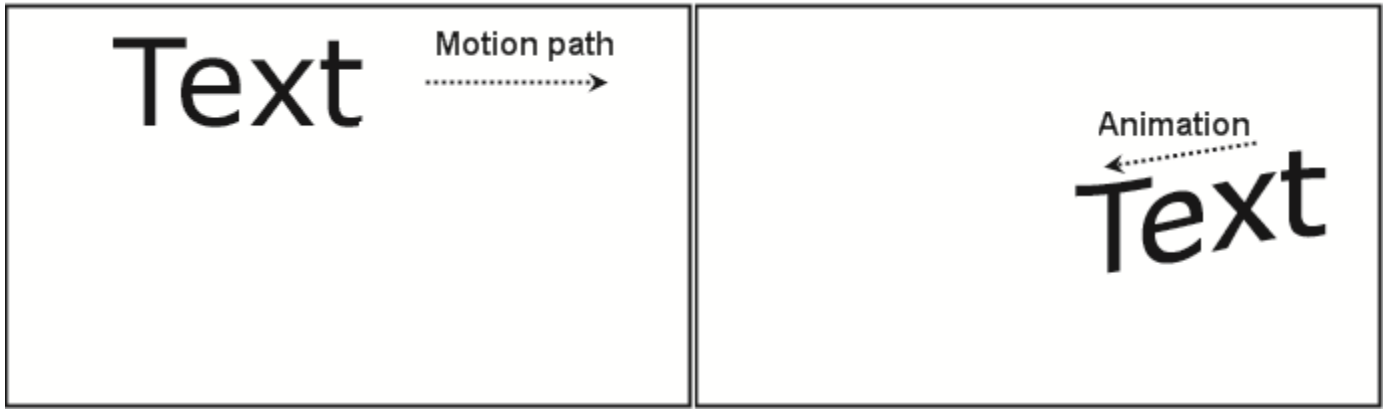
Lossy compression

- A representation of the original uncompressed image is preserved.
- The image appears to be a copy of the original image, but in real it is not a copy.
- Lossy compression achieves smaller file sizes when compared with lossless compression.
- Lossy compression algorithms allow variable compression that comprises on image quality for file size.

Graphic Format 3-4

➢ Animation

- Some graphic format consists of a series of frames that are played one after the other giving an impression of animation.
- Following figure shows an animated graphic:

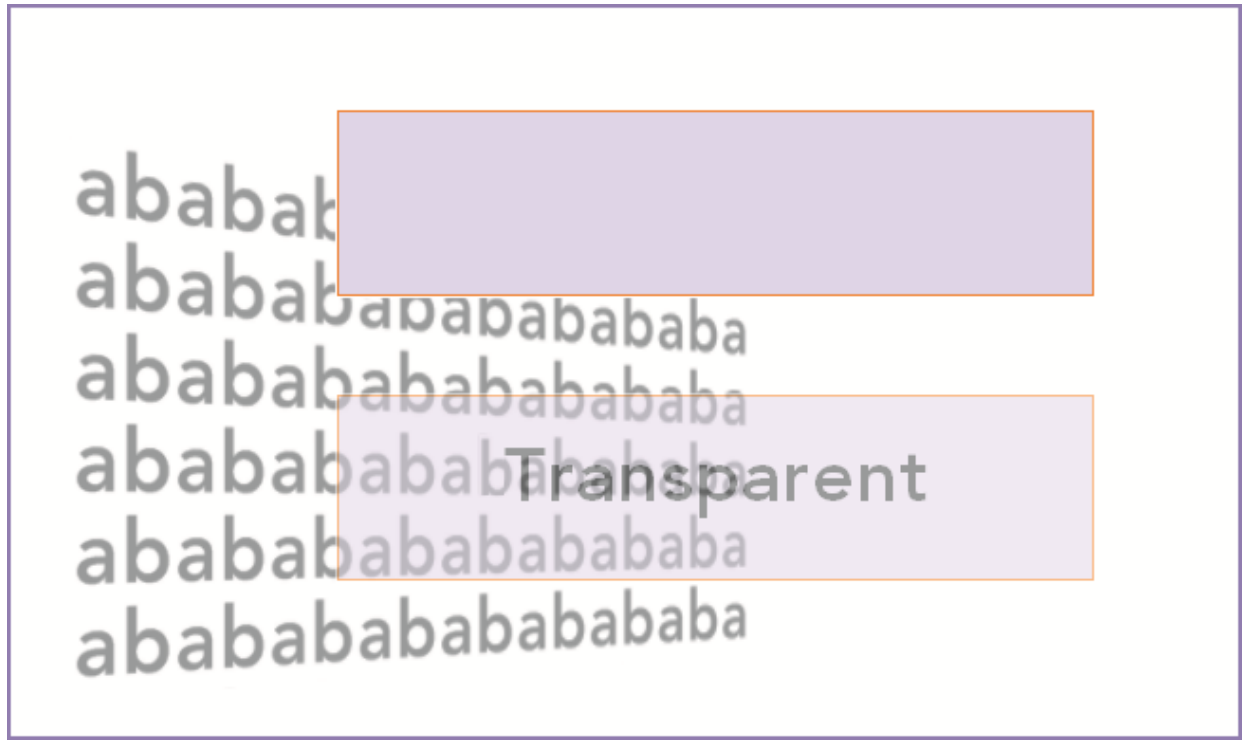


➢ Transparency

- It is very common on the Web to display an image on a Web page that appears directly against the background color of the page.
- The background color of the Web page shows through the transparent portion of the image.

Graphic Format 4-4

- In a transparent image, one and only one color can be hidden.
- If the color chosen to make transparent is same as the background of the inserted image, then an irregularly shaped image appears to float on the page.
- Following figure shows a transparent image:



Graphic Format for the Web 1-2

- JPEG and PNG graphics provide maximum compatibility with all the devices accessing Web pages.
- For photos, use of JPEG graphic format and for screen-shots and drawings use of PNG graphic format is recommended.

➢ JPEG

- Uses a lossy compression which means that image quality is lost in the process of compressing the image.
- For continuous tone pictures such as photos, JPEG should be used.
- Most JPEG editors allow user to specify amount of detail that the user is prepared to lose.
- If the quality is reduced, then loss is visible; JPEG is about half the size of PNG.

➢ PNG

- Uses lossless compression, which means there is no loss of any image detail.
- Designed for transferring images on the Internet and not for professional-quality print graphics.
- Therefore, it does not support non-RGB color spaces such as CMYK.
- Supports high color and partial transparency using alpha channels.

Graphic Format for the Web 2-2

➢ GIF

- Uses a lossless compression which means that there is no loss in quality when the image is compressed.
- The uncompressed image stores its information in a linear fashion.
- Each line of pixels is read from left to right.
- An interlaced GIF file stores the lines of the image in a different order.
- Animated graphics are stored in GIF format.

Compatibility and appearance are the keywords on the Web.

The inserted images must be visible and undistorted when appearing on any recipient's device.

The Web designer can make assumptions that the Website will open in a computer which will have minimum resolution of 800x600 pixel display capability.

If a mobile based Web page needs to be created then the specifications will change.

Graphic Insertion 1-6

The `IMG` element is an empty element, which allows the user to insert an image in a Web page.

It allows insertion of images and diagrams.

The commonly used graphic formats that are supported are namely, GIF, JPEG, BITMAP (BMP), and PNG.

The `<img>` tag reserves a space for the image and does not insert the image in the HTML page.

It creates a link between the image and the HTML page.

Graphic Insertion 2-6

- Following table lists the commonly used attributes of the `IMG` element:

Data Type	Description
`src`	Specifies the path of an image that is to be displayed.
`height`	Specifies the height of an image.
`width`	Specifies the width of an image.

- Following Code Snippet demonstrates displaying an image using the `IMG` element:

```
<body>
<img src="UNO.jpg" width="225" height="151"/>
</body>
```

- The code uses the `src` attribute of the `IMG` element to insert a `JPEG` image.
- The attribute specifies the name of the image and also indicates that the image is present in the same folder where the HTML file is saved.
- The width and height of the image is set to `225` and `151` pixels respectively by using the `width` and `height` attribute.
- A pixel refers to the smallest dot on the monitor screen.

Graphic Insertion 3-6

- An image can also be stored in a subfolder of the folder containing the HTML file.
- In such cases, a reference to the image is made by using the sub folder name as shown in the following Code Snippet:

```
<body>
<img src="image_folder/UNO.jpg" width="225" height="151"/>
</body>
```

- To align the image, the `float` style attribute can be used to specify the inline style for the element.
- This will force the image to be aligned to the left or right side of the screen and wrap the surrounding text around the image.
- Following Code Snippet demonstrates the use of the float style:

```
<body>
<img src="image_folder/UNO.jpg" style="float:left"/>
</body>
```

Graphic Insertion 4-6

- Following table lists the values of `float` property in the `<img>` tag:

Value	Description
left	The element floats to the left.
right	The element floats to the right.
none	The element does not float and is the default value.
inherit	The element specifies that the value of the float property should be inherited from the parent element.

HTML5 introduced a new `<figure>` tag that acts as a container containing the `<img>` tag.

It is not a replacement for `<img>` tag, but acts as a container into which the `<img>` tag is placed.

The `<figure>` tag specifies self-contained content, such as illustrations, diagrams, photos, code listings, and so on.

The content of the `<figure>` element is related to the main flow, its position is independent of the main flow.

Graphic Insertion 5-6

- Following Code Snippet demonstrates the use of `<figure>` tag:

```
<figure>
  <img src="logo.gif" style="float: left" width="304"
   height="228" />
</figure>
```

- The main advantage of using `<figure>` tag is that it allows the user to use `<figcaption>` tag along with it.
- The `<figcaption>` tag allows the user to add a caption to the image.
- The caption always appears along with the image even if the image floats in Web site layout.
- Following Code Snippet demonstrates the use of `<figcaption>` tag:

```
<figure>
<img src="logo.gif">
<figcaption>This diagram shows the logo of a
  product.</figcaption>
</figure>
```

Graphic Insertion 6-6

- The `<figure>` tag can also assign styles and other attributes to the `<figure>` element using an external or internal style sheet.
- A single caption to a group of images can be added using the `<figure>` tag.
- Following Code Snippet demonstrates how to assign a single caption to a group of images:

```
<figure>
<img src="flower1.jpg">
<img src="flower2.jpg">
<img src="flower3.jpg">
<figcaption>The different types of flowers</figcaption>
</figure>
```

- Following figure shows output of a single caption to a group of images:

CSS Image Sizing and Padding 1-3

- Size of an image is specified in pixels.
- The `height` and `width` property sets the height and width of the image.
- One can specify the width and the height will be resized or vice versa.
- Following Code Snippet demonstrates CSS code for setting the image height and width property:

```
p.ex
{
height:100px;
width:100px;
}
```

CSS Image Sizing and Padding 2-3

- Following table lists different CSS properties and values of images:

Property	Description	Values
height	Sets the height of an element	• Auto • Length • % • inherit
max-height	Sets the maximum height of an element	• none • length • % • inherit
max-width	Sets the maximum width of an element	• none • length • % • Inherit
min-height	Sets the minimum height of an element	• length • % • Inherit
min-width	Sets the minimum width of an element	• length • % • Inherit
width	Sets the width of an element	• auto • length • % • inherit

CSS Image Sizing and Padding 3-3

- Following table lists various values used with height and width properties:

Value	Description
auto	The browser calculates the height and is the default value
height	Defines the height in pixels (px) % Defines the height of the containing block in percent format
width	Defines the width in pixels (px) % Defines the width of the containing block in percent format
inherit	Specifies that the value of the property should be inherited from the parent element

Padding 1-3

- The CSS `padding` property is used to specify the space between the element border and the element content.
- The background color of the element affects the padding property.
- Using separate properties such as top, right, bottom, and left, different padding values can be specified and the padding can be changed separately.
- Following table lists various values used in `padding` property:

Value	Description
length	This property specifies a fixed value for padding in pixels, pt, em, and so on.
%	This property specifies a value for padding in % of the containing element.

Padding 2-3

- Following Code Snippet demonstrates the CSS code used for specifying different padding values for different sides:

```
padding-top:10px;
padding-bottom:10px;
padding-right:15px;
padding-left:15px;
```

- Instead of using different padding for different sides, users can use a shorthand property.
- A shorthand property is one where all the padding properties for different sides are specified in one property.
- The shorthand property for all the padding properties is `padding`.
- Following Code Snippet demonstrates the use of the shorthand property for padding:

```
padding:25px 50px 75px 100px;
```

where,

top padding is `25px`, right padding is `50px`, bottom padding is `75px`, and left padding is `100px`.

Padding 3-3

- Following table lists all CSS padding properties:

Property	Description
padding	The browser calculates the height and is the default value
padding-bottom	Defines the length in pixels (px)
padding-left	Defines the height of the containing block in percent format
padding-right	Specifies that the value of the property should be inherited from the parent element
padding-top	Sets the top padding of an element

Thumbnail Graphics 1-4

- A thumbnail is a small image or a part of a larger image.
- Clicking the thumbnail image will link to the larger original image, which can be viewed and downloaded. Even a hover effect can be given through CSS and JavaScript.

Following Code Snippet demonstrates an HTML code for inclusion of a thumbnail image:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <title>Thumbnail</title>
    <style>
          body {
                    margin:0;
                    padding:40px 80px;
                    background:#fff;
                    font:70% Arial, Helvetica, sans-serif;
                    color:#555;
                    line-height:100%;
               }
h1, h2{
                    font-size:180%;
                    font-weight:normal;
                    color:#555;
          }
          p{
                    margin:1em 0;
          }
          p.text{
                    width:500px;
          }
          a{
                    color:#f20;
                    text-decoration:none;
          }
```

Thumbnail Graphics 2-4

```
a:hover{
            color:#999;
        }
img{
            border:none;
   }
/* thumbnail list */
ul#thumbs, ul#thumbs li{
                  margin:0;
                  padding:0;
                        list-style:none;
             }
ul#thumbs li{
                        float:left;
                        margin-right:0px;
                        border:1px solid #999;
                        padding:2px;
            }
ul#thumbs a{
                        display:block;
                        float:left;
                        width:125px;
                        height:135px;
                        line-height:50px;
                        overflow:hidden;
                        position:relative;
                        z-index:1;
              }
ul#thumbs a img{
                        float:left;
                        position:absolute;
                        top:0px;
                        left:0px;
                  }
```

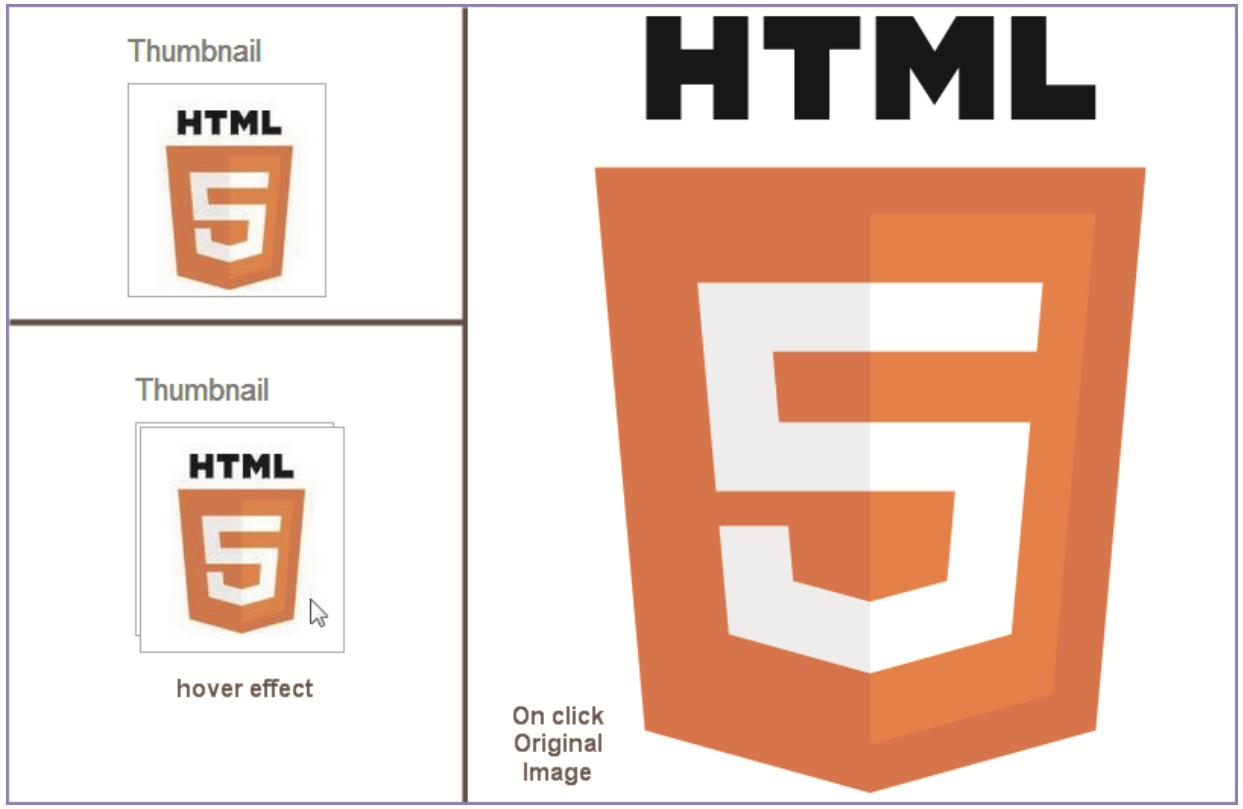
Thumbnail Graphics 3-4

```
/* mouse over */
ul#thumbs a:hover{
                    overflow:visible;
                     z-index:1000;
                    border:none;
                  }
ul#thumbs a:hover img{
                     border:1px solid #999;
                     background:#fff;
                     padding:2px;
                  }
/* mouse over */
/* clearing floats */
ul#thumbs:after, li#thumbs:after{
                     content:".";
                     display:block;
                     height:0;
                     clear:both;
                     visibility:hidden;
                     }
ul#thumbs, li#thumbs{
                     display:block;
                       }
ul#thumbs, li#thumbs{
                     min-height:1%;
                     }
* html ul#thumbs, * html li#thumbs{
                     height:1%;
                     }
```

```
/* clearing floats */
/* thumbnail list */
</style>
  </head>
   <body>
      <h2>Thumbnail</h2>
       <ul id="thumbs">
       <li><a href="HTML5.png"
target="_blank">
           <img
src="HTML5_small.jpg"/></a></li>
         </ul>
  </body>
</html>
```

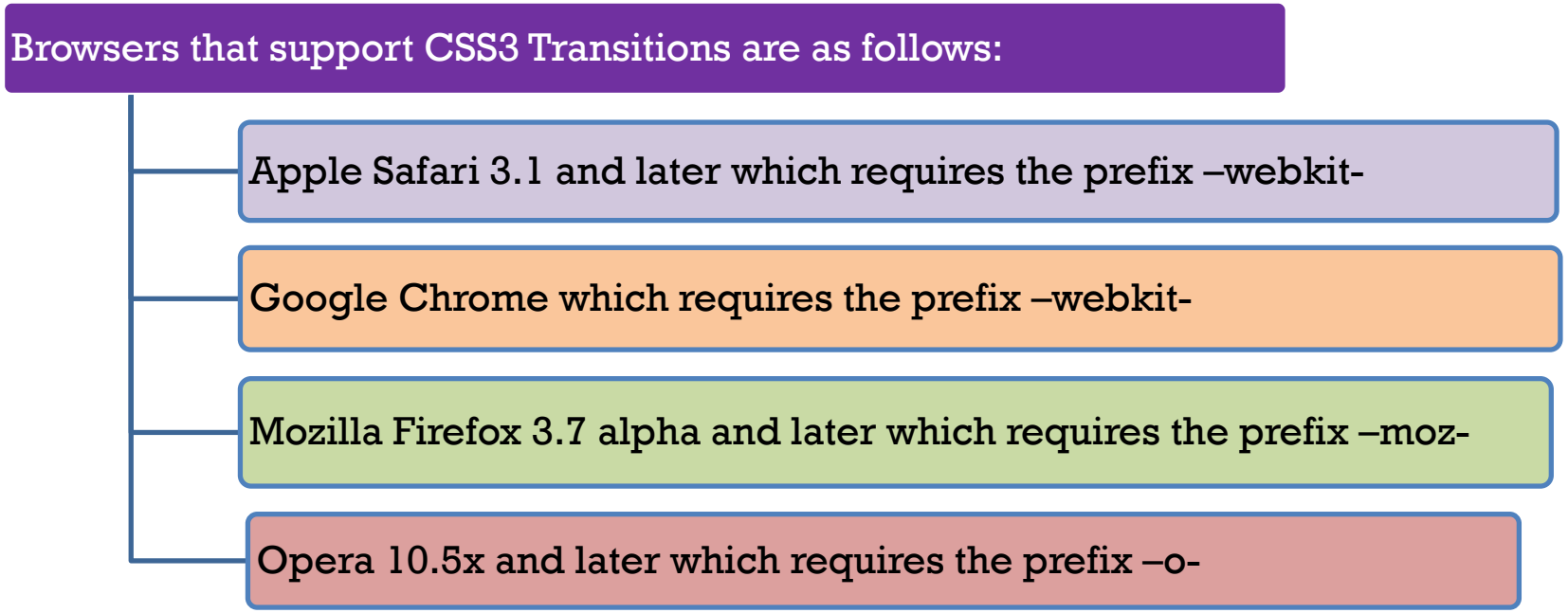
Thumbnail Graphics 4-4

- Following figure shows output of thumbnail with hover effect:

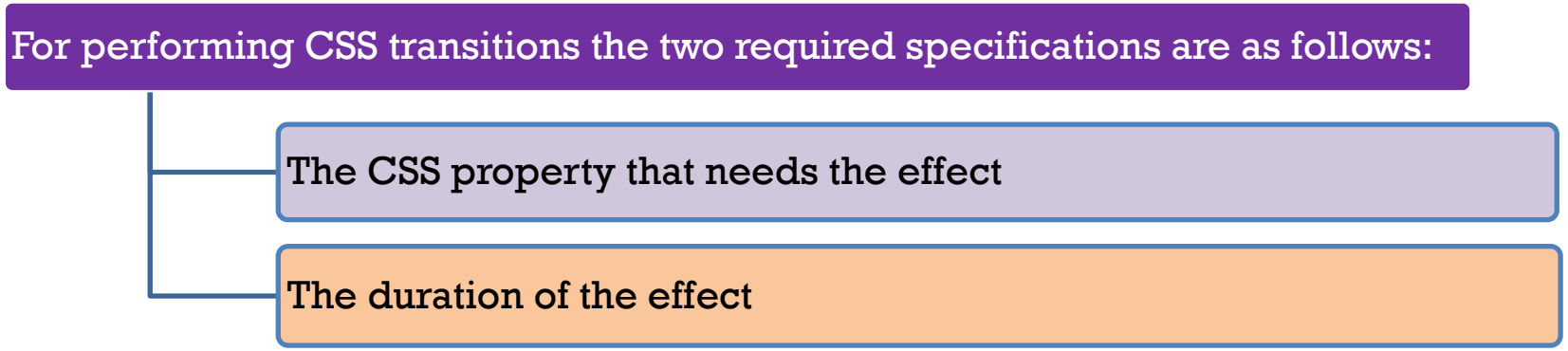


Working with CSS3 Transitions 1-6

- In 2007, Apple introduced the CSS transition, which later became a proprietary feature of Safari called CSS Animation.
- Representatives from Apple and Mozilla began adding the CSS transitions module to the CSS Level 3 specification, closely modeled on what Apple had already added to Webkit and moz.



Working with CSS3 Transitions 2-6



- Following Code Snippet demonstrates the use of transition effect on the `width` property for three seconds:

```
div
{
    transition: width 3s;
    -moz-transition: width 3s; /* Firefox 4 */
    -webkit-transition: width 3s; /* Safari and Chrome */
    -o-transition: width 3s; /* Opera */
}
```

- The effect will start when the specified CSS property changes value.
- The CSS property changes its value typically when a user moves a mouse over an element.
- Thus, the user can set the hover for `<div>` elements.

Working with CSS3 Transitions 3-6

- Following Code Snippet demonstrates setting the hover for elements:

```
div:hover
{
  width:200px;
}
```

- Following table lists all the transition properties:

Property	Description
transition	Is a shorthand property and is used for setting the four transition properties into a single property.
transition-property	Is used for specifying the name of the CSS property for which the transition value is set.
transition-duration	Is used for defining the duration of the transition. Default value is 0.
transition-timing-function	Is used for describing how the speed during a transition will be calculated. Default value is 'ease'.
transition-delay	Is used for defining the start of the transition. Default value is 0.

Working with CSS3 Transitions 4-6

- Following Code Snippet demonstrates an HTML and CSS code using all transition properties:

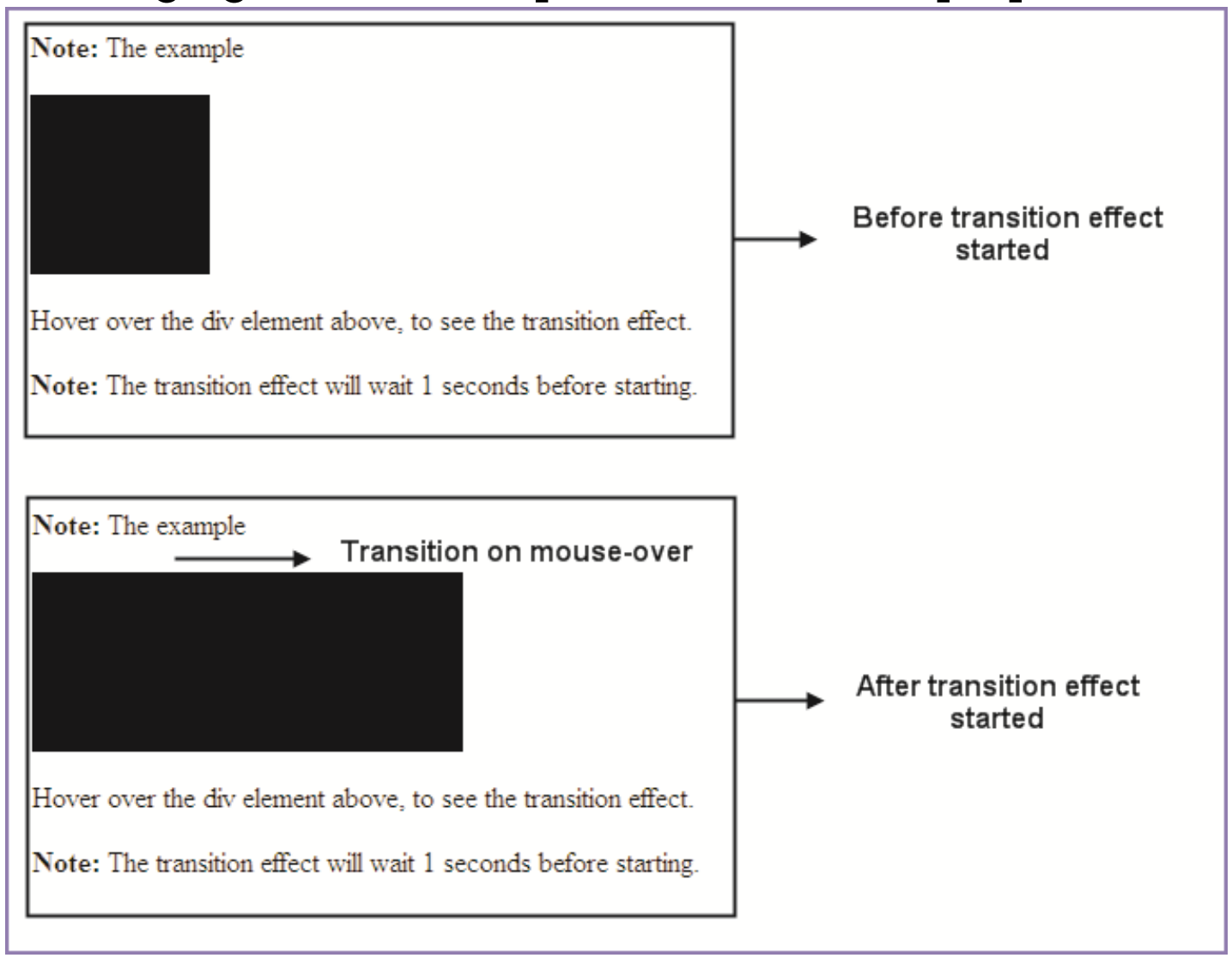
```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:#000000;
transition-property:width;
transition-duration:2s;
transition-timing-function:linear;
transition-delay:1s;
/* Firefox 4 */
-moz-transition-property:width;
-moz-transition-duration:2s;
-moz-transition-timing-function:linear;
-moz-transition-delay:1s;
```

Working with CSS3 Transitions 5-6

```
/* Safari and Chrome */
-webkit-transition-property:width;
-webkit-transition-duration:2s;
-webkit-transition-timing-function:linear;
-webkit-transition-delay:1s;
}
div:hover
{
width:500px;
}
</style> </head>
<body>
<p><b>Note:</b> The example</p>

<div></div>

<p>Hover over the div element above, to see the transition
   effect.</p>
<p><b>Note:</b> The transition effect will wait 1 seconds
     before starting.</p>
</body></html>
```

Working with CSS3 Transitions 6-6

- Following figure shows output of all transition properties:



CSS3 Animation

- CSS3 animations can animate transitions of one CSS style configuration to another.

Two components of animation are as follows:

- An animation style describing the animation.
- A keyframes set that specifies the start and end states of the animation's CSS style and possible intermediate waypoints along the way.

Advantages of CSS3 animations over script-based animation techniques are as follows:

- Easy to use and anybody can create them without the knowledge of JavaScript.
- Executes well even under reasonable system load.
- Allows the browser to control the animation sequence, optimize performance, and efficiency.

Configuring the Animation 1-7

- A CSS animation sequence can be created by styling the element with the animation property.
- This property can be used to configure the timing, duration, and sequence of the animation.
- `@keyframes` rule defines the appearance of the animation.
- The keyframe is used to describe the rendering of the element in the animation sequence.
- Following table lists the `@keyframes` rule and all the animation properties:

Property	Description
`@keyframes`	Is used for specifying the animation.
`animation`	Is a shorthand property representing all the animation properties, except the animation-play-state property.
`animation-name`	Is used for specifying the name of the @keyframes animation.
`animation-duration`	Is used for specifying the duration of an animation cycle in seconds or milliseconds. Default value is 0.
`animation-timing-function`	Is used for describing the progress of animation over one cycle of its duration. Default value is 'ease'.

Configuring the Animation 2-7

Property	Description
animation-delay	Is used for specifying the start value of animation. Default value is 0.
animation-iteration-count	Is used for specifying the number of times an animation is played. Default value is 1.
animation-direction	Is used for specifying whether or not the animation should play in reverse on alternate cycles. Default value is 'normal'.
animation-play-state	Is used for specifying the state of the animation, that is whether it is running or paused. Default value is 'running'.

- The syntax for `@keyframes` is as follows:

Syntax:

```
@keyframes myfirst
{
from {background: red;}
to {background: yellow;}
}
@-moz-keyframes myfirst /* Firefox */
{
from {background: red;}
to {background: yellow;}
}
@-webkit-keyframes myfirst /* Safari and Chrome */
{
from {background: red;}
to {background: yellow;}
}
```

Configuring the Animation 3-7

Following Code Snippet demonstrates HTML and CSS code of `@keyframes` rule and all the animation properties:

```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div {
width:200px;
height:200px;
background:red;
position:relative;
border-radius:100px;
animation-name:myfirst;
animation-duration:4s;
animation-timing-function:linear;
animation-delay:1s;
animation-iteration-count:infinite;
animation-direction:alternate;
animation-play-state:running;
/* Firefox: */
-moz-border-radius:100px;
-moz-animation-name:myfirst;
-moz-animation-duration:4s;
-moz-animation-timing-function:linear;
-moz-animation-delay:1s;
-moz-animation-iteration-count:infinite;
-moz-animation-direction:alternate;
-moz-animation-play-state:running;
```

Configuring the Animation 5-7

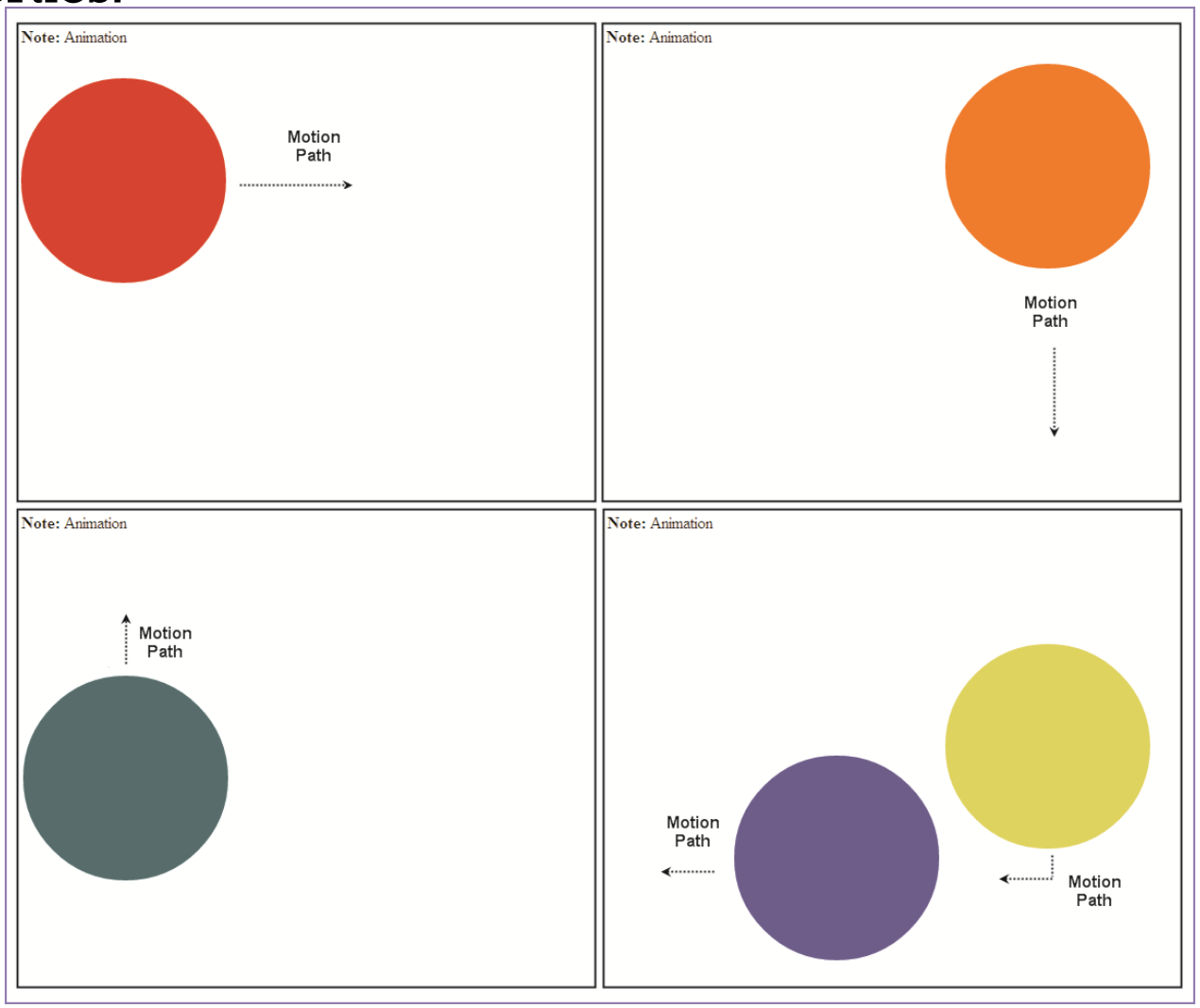
```
/* Safari and Chrome: */
-webkit-border-radius:100px;
-webkit-animation-name:myfirst;
-webkit-animation-duration:4s;
-webkit-animation-timing-function:linear;
-webkit-animation-delay:1s;
-webkit-animation-iteration-count:infinite;
-webkit-animation-direction:alternate;
-webkit-animation-play-state:running;
}
@keyframes myfirst
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:300px; top:0px;}
50%  {background:blue; left:300px; top:300px;}
75%  {background:green; left:0px; top:300px;}
100% {background:red; left:0px; top:0px;}
@-moz-keyframes myfirst /* Firefox */
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:300px; top:0px;}

}
```

Configuring the Animation 6-7

```
50%  {background:blue; left:300px; top:300px;}
75%  {background:green; left:0px; top:300px;}
100% {background:red; left:0px; top:0px;}
}
@-webkit-keyframes myfirst /* Safari and Chrome */
{
0%   {background:red; left:0px; top:0px;}
25%  {background:yellow; left:200px; top:0px;}
50%  {background:blue; left:200px; top:200px;}
75%  {background:green; left:0px; top:200px;}
100% {background:red; left:0px; top:0px;}
}
</style>
</head>
<body>
<p><b>Note:</b> Animation</p>
<div></div>
</body>
</html>
```

Configuring the Animation 7-7

- Following figure shows the output of `@keyframes` rule and all the animation properties:



Using CSS3 on Mobile Devices 1-2

There are different ways to provide Web pages for mobile devices.

The user can make use of style sheet for the handheld devices (all mobile browsers do not recognize it).

IPhone's Safari and Opera's Mini browsers support a new feature of CSS3 called media queries.

These queries allow the user to specify a conditional expression for media type.

- Following Code Snippet shows the use of a conditional expression for displaying a link element where the maximum screen width for mobile devices is 480 pixels:

```
<link rel="stylesheet" href="styles/mobile.css" media="only
 screen and (max-device-width: 480px)"/>
```

- The user can also specify another link element for screen media with a minimum screen width of 481 pixels.
- In other words, the style sheet for this element can be used for standard computer screens.

Using CSS3 on Mobile Devices 2-2

- Most mobile Websites are created to precede the domain name of the main site with m for example **m.aptech-education.com**.
- To detect a mobile device, a Web site can use JavaScript on the client, a scripting language on the server, or Wireless Universal Resource File (WURFL) on the server.

Five ways to provide Web pages for mobile devices are as follows:

- Define a style sheet for mobile devices
- Include a link to a mobile version of the Website
- Use JavaScript to detect mobile devices and redirect
- Use a server-side scripting language to detect and redirect
- Use the WURFL to detect mobile devices

Coding for Optimum Browser Compatibility 1-3

Web browser compatibility measures are undertaken to provide predictability and consistency across the preferable Web browsers of the targeted end users.

Cross browser compatibility means a Website that is attuned and reliable in looks, layout, color, functionality, interactivity, and proportion.

Cross browser compatibility is across all existing Web browsers, regardless of the browsers' insignificance or popularity differences from version to version.

Multi-browser compatibility is constant and it is functionally rendered across the most commonly used browsers in a client's target market.

HTML5 uses different standards and is supported by various browsers. These browsers provide different version of support.

Rendering engines are a set of tools that are used in most browsers that supports different HTML features.

Coding for Optimum Browser Compatibility 2-3

- Some of the rendering engines of different browsers are as follows:

Gecko

- The Gecko engine is the main engine of Mozilla Firefox and a number of related browsers.
- It has support for various HTML5 features.

Trident

- The Trident engine is used by different versions of Internet Explorer (IE).
- Currently, HTML5 is not majorly supported by the Trident engine.

WebKit

- The WebKit engines is supported mainly for the Safari browser used in Apple Macs, iPhones, iPads, and other Apple products.
- This engine is based on the open source KHTML project.

Presto

- Presto is the engine used in the Opera browsers.
- Opera browsers are considered to be a technically superior browser, but market share of Opera browsers is still low.

Coding for Optimum Browser Compatibility 3-3

- Best practices for optimum browser compatibility are as follows:

Test the Website in different browsers

- Review the Web site's appearance and functionality on multiple browsers to ensure that all the users are getting the same experience according to the design.
- Preferably test on different versions of the same browser also as they can show the Website differently.

Write a good clean HTML code

- To ensure that the page looks same in all browsers is to write Web pages using valid HTML and CSS codes and then, test it in many browsers.
- Using External CSS can help pages render and load faster.

Summary

- The text styles specify and control the appearance of the text in a Web page.
- Indenting is the process of offsetting text from its normal position, either to the left or to the right.
- CSS border property specifies the style, color, and width of the border.
- The border-color property accepts different color values that determine the different shades of color to be applied to the borders.
- The values of the different border properties determine the type of effect to be applied to the borders.
- In CSS, the text-align property is used for horizontal alignment of text in an element.
- In CSS, the line-height property is used for vertical alignment of text in an element.