Displaying Graphics & CSS3 Animation

Lokesh Kumar

Introduction

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

Graphic Format

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 the color depth indicates higher range of colors used.

Compression File/Size

➔ **Compression/file size** – Graphic files are large, so images are compressed using various techniques. 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. In some compression formats, images with less complexity results in smaller compressed file sizes.

The two types of image file compression algorithms used are as follows:

- **Lossless compression** – In this algorithm, file size is reduced but preserves a copy of the original uncompressed image. Lossless compression avoids accumulating stages of re-compression when editing images.
- **Lossy compression** – In this algorithm, a representation of the original uncompressed image is preserved. The image appears to be a copy of the original image but in actuality it is not a copy. Lossy compression achieves smaller file sizes when compared with lossless compression. Generally, lossy compression algorithms allow variable compression that comprises on image quality for file size.

Animation

- **Animation** – Some graphic format consists of a series of frames that are played one after the other giving an impression of animation. Animated graphics are typically used on a Web page to attract visitor's attention.

Transparency

- **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. 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. Figure 7.2 shows a transparent image.

Graphic Format For Web

For Web pages, use of JPEG and PNG graphics are recommended as it provides maximum compatibility with all the devices that might be accessing the Web page. For photos, use of JPEG graphic format and for screen-shots and drawings use of PNG graphic format is recommended. Both these formats compress the picture information to reduce the download time and increase the downloading speed.

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

Graphic Format (contd.)

- **PNG** - It uses lossless compression, which means there is no loss of any image detail. PNG was 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. It supports high color and partial transparency using alpha channels.

- **GIF** - It 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.

Graphic Insertion

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.

Table 7.1 lists the commonly used attributes of the IMG element.

Attributes	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.

Sample 1

```
<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.

Sample 2

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 Code Snippet 2.

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

Sample 3

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. Code Snippet 3 demonstrates the use of the float style.

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

Values of FLOAT Property

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.

Table 7.2: Values of Float Property in <img> Tag

Figure Tag

HTML5 introduced a new <figure> tag. The <figure> tag acts as a container containing the <img> tag. In other words, 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.

While the content of the <figure> element is related to the main flow, its position is independent of the main flow, and if removed it does not affect the flow of the document.

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

Figcaption

The main advantage of using <figure> tag is that it allows the user to use the <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.

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

Figure

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.

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

Sample

The different types of flowers

CSS Image Sizing and Padding

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.

Note - The height and width property does not include padding, borders, or margins.

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

Property	Description	Values
height	Sets the height of an element	• Auto • Length • % • inherit

Properties and their Values

Property	Description	Values
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

Different Values

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

Padding

The CSS padding property is used to specify the space between the element border and the element content. It is used to separate them from the surrounding element. 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.

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 Sample

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

In the code, the value for padding was set for all the sides.

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 the different sides are specified in one property. This will result in a shortened code.

Shorthand Property

The shorthand property for all the padding properties is `padding`. The property can be used to specify one to four values for each of the side. Code Snippet 9 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`.

Thumbnail Graphics

The speed of loading a page of a Web site is reduced if high-resolution graphics are used. High-resolution graphics are required to improve the effectiveness of the site and cannot be avoided. Hence, to avoid this issue, thumbnails are used.

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.

Example

```
<!DOCTYPE html>
<html lang="en">
 <head>
  <meta charset="utf-8">
  <title>Thumbnail</title>
  <style>
/* general */
      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
}
a:hover{
            color:#999;
               }
img{
            border:none;
}
        /* // general */
        /* 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;
            }
```

Example Contd.

```
ul#thumbs a img{
      float:left;
      position:absolute;
      top:0px;
      left:0px;
}
/* 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>
```

Output


Thumbnail
HTML
Thumbnail
HTML
hover effect
HTML
On click
Original
Image

CSS3 Transitions

Interactivity is one of the important aspects of animation. Earlier, a combination of HTML, CSS, and JavaScript were used to animate objects on the Web. 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.

All the browsers do not support CSS3 transitions. Browsers that support CSS3 Transitions are as follows:

- Apple Safari 3.1 and later which requires the prefix –webkit-
- Google Chrome which requires the prefix –webkit-
- Mozilla Firefox 3.7 alpha and later which requires the prefix –moz-
- Opera 10.5x and later which requires the prefix –o-

CSS3 Tansitions (contd.)

At the moment Internet Explorer 9 does not support CSS3 Transitions.

For performing CSS transitions the two required specifications are as follows:

- ➔ The CSS property that needs the effect
- ➔ The duration of the effect

Sample

```
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 elements. Code Snippet 12 demonstrates the same.

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

Property

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

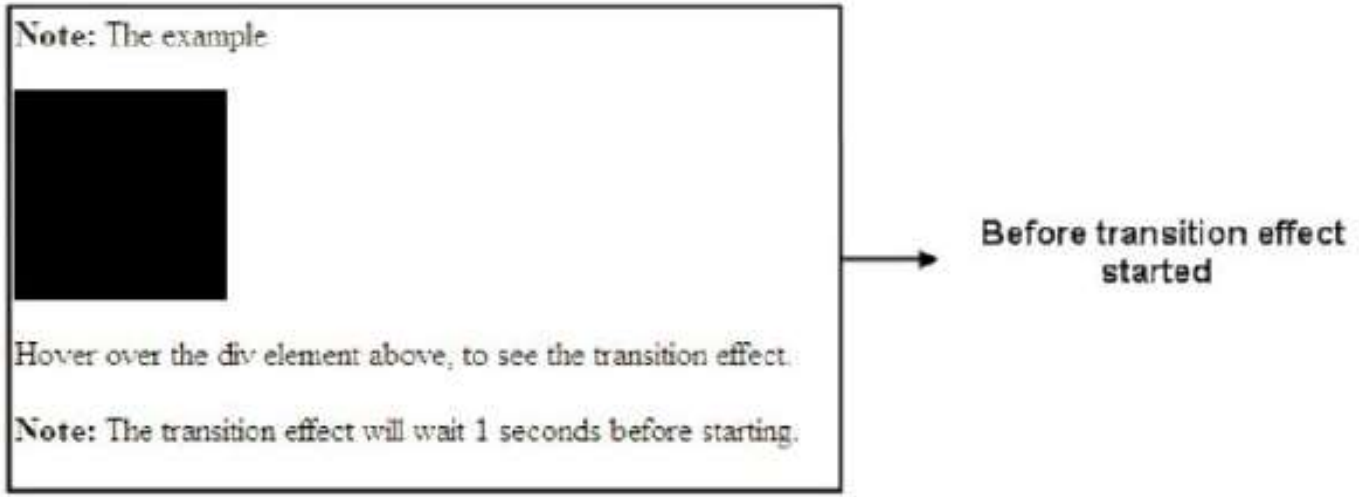
Example

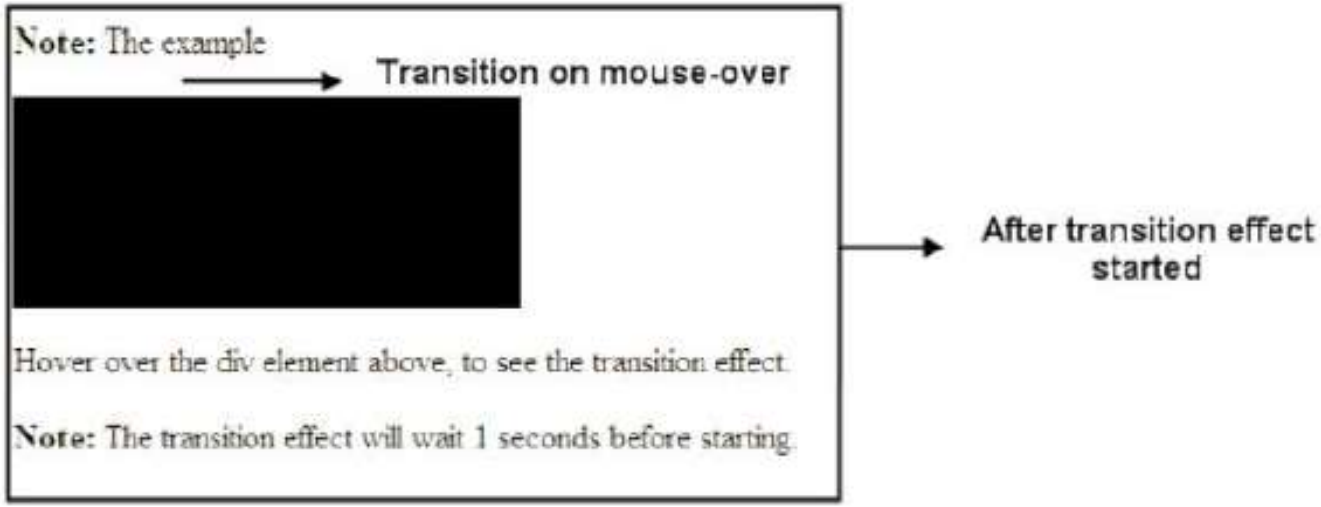
```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:100px;
height:100px;
background:#000000;
transition-property:width;
transition-duration:2s;
transition-timing-function:linear;
transition-delay:1s;
/* Firefox 4 */
-moz-transition-property:width;
-moz-transition-duration:2s;
-moz-transition-timing-function:linear;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

Example (contd.)

```
<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>
```

Output


Note: The example
Hover over the div element above, to see the transition effect.
Note: The transition effect will wait 1 seconds before starting.
Before transition effect started


Note: The example
Transition on mouse-over
Hover over the div element above, to see the transition effect.
Note: The transition effect will wait 1 seconds before starting.
After transition effect started

CSS3 Animation

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

The three advantages to CSS3 animations over script-based animation techniques are as follows:

1. Easy to use and anybody can create them without the knowledge of JavaScript.
2. Executes well even under reasonable system load. As simple animations perform poorly in JavaScript, the rendering engine uses the frame-skipping techniques to allow smooth flow of animation.
3. Allows the browser to control the animation sequence, optimize performance and efficiency by reducing the update frequency of animations executing in tabs that aren't currently visible.

Configuring Animation

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 define the appearance of the animation. The keyframe is used to describe the rendering of the element in the animation sequence.

Property

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"
`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"

Keyframes

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;}
}
```

Explainaton

The animation created using `@keyframes` must be bound with the selector for effective execution. For this, specify the name of the animation and the duration of the animation to the selector.

Code Snippet 14 demonstrates HTML and CSS code of `@keyframes` rule and all the animation properties.

Sample

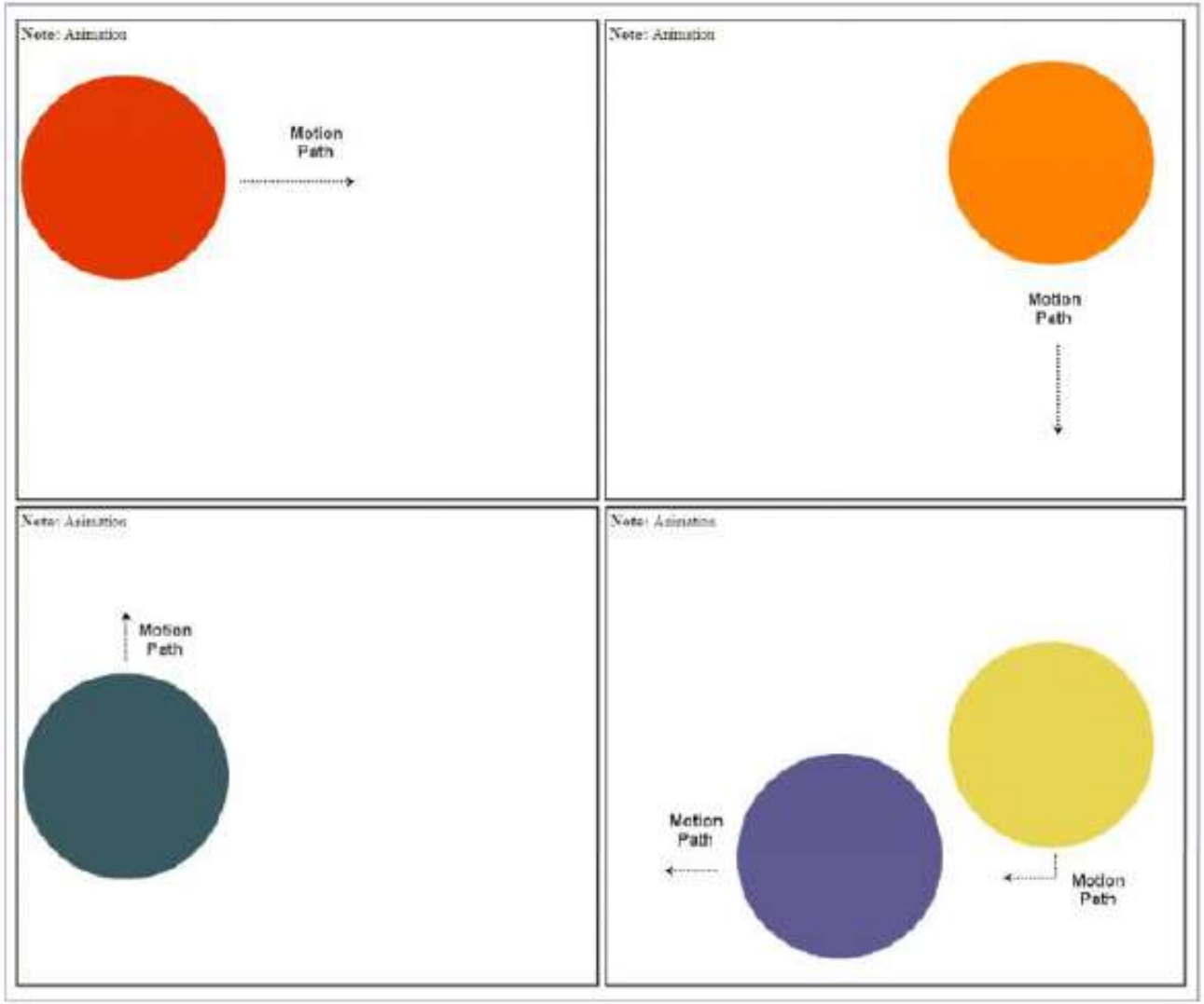
```
<!DOCTYPE html>
<html>
<head>
<style type="text/css">
div
{
width:200px;
height:200px;
background:red;
position:relative;
border-radius:100px;
animation-name:myfirst;
animation-duration:4s;
animation-timing-function:linear;
animation-delay:1s;
animation-iteration-count:infinite;
animation-direction:alternate;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

Sample (contd.)

```
@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;}
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>
```

Output


Note: Animation
Motion Path
Note: Animation
Motion Path
Note: Animation
Motion Path
Note: Animation
Motion Path
Motion Path