Session: 17

Canvas and JavaScript

Objectives

- Describe Canvas in HTML5
- Explain the procedure to draw lines
- Explain the procedure to use color and transparency
- Explain the procedure to work with various drawing objects
- Describe working with images and text
- Describe the procedure to create Web page events with JavaScript and jQuery
- Describe the process of including external content in Web pages

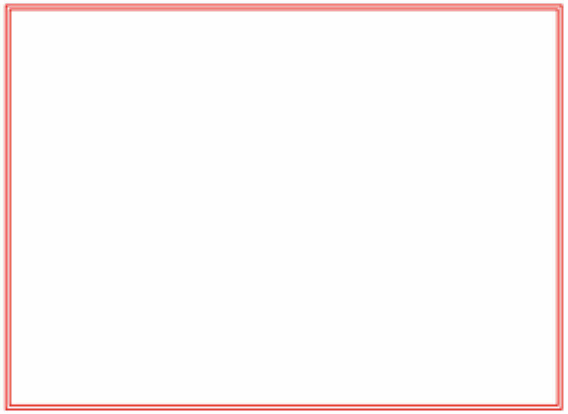
Canvas Element 1-3

- can be used to draw shapes on Web sites as well as to dynamically draw graphics using JavaScript.
- is represented like a rectangle on a page and allows the user to draw arcs, text, shapes, gradients, and patterns.
- is like the <div>, <table>, or <a> tag except that the content used in it is rendered through JavaScript.
- does not contain any drawing abilities, instead, the drawing is done using a JavaScript code.
- Using <canvas> with JavaScript improves the overall performance of Web sites and avoids the requirement to download images from the sites.

Canvas Element 2-3

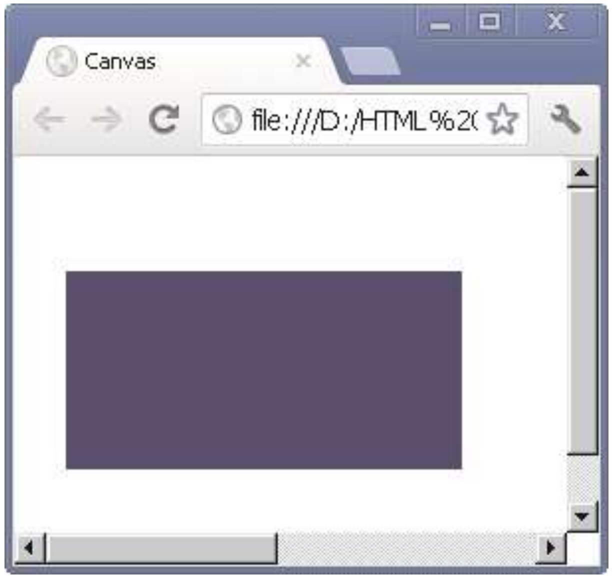
- The Code Snippet demonstrates the use of <canvas> element.

```
<!DOCTYPE HTML>
 <html>
  <head>
   <title> Canvas </title>
    <style>
       canvas{border: medium double red; margin: 4px}
   </style>
      </head>
  <body>
    <canvas width="278" height="200">
    </canvas>
  </body>
 </html>
```



Canvas Element 3-3

- The <canvas> element in DOM exposes the HTMLCanvasElement interface.
- This interface provides the methods and properties for changing the presentation and layout of canvas elements.
- The HTMLCanvasElement has a getContext(context) method that returns the drawing context for the canvas.



Drawing a Line in Canvas

- You can create lines in a canvas using the stroke(), beginPath(), lineTo(), and moveTo() methods.
- Syntax to create a line in canvas:

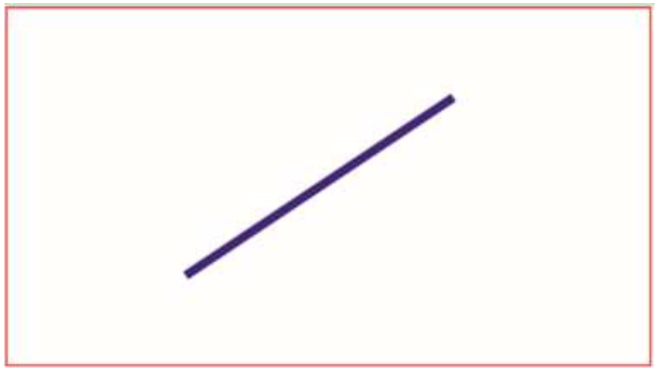
Syntax:

```
ctext.beginPath();

ctext.moveTo(x,y);

ctext.lineTo(x,y);

ctext.stroke();
```



where,

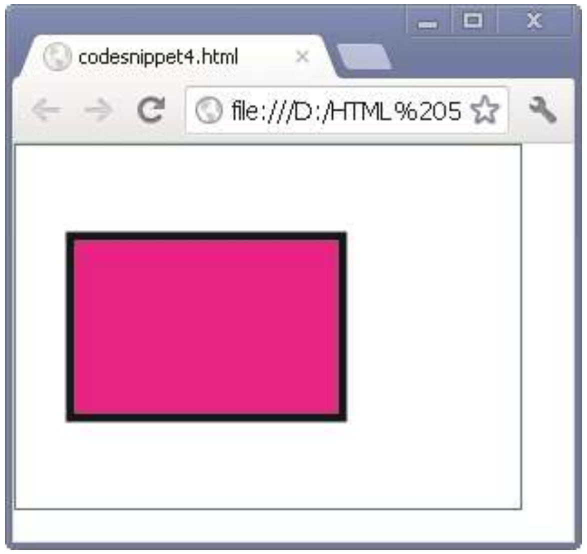
- ctext - specifies a context object
- beginPath() - Specifies a new drawing path
- moveTo() - Specifies the creation of new sub path to the given position
- lineTo() - Specifies the drawing of a line from the context position to the given position
- stroke() - Specifies how to assign a color to the line and display it

Working with Drawing Objects 1-5

➢ Rectangle

- With HTML5 canvas, can create a rectangle using the **rect()** method.
- The HTML5 canvas is placed by using the x and y parameters and appropriately sized through height and width properties.

Properties and Methods
`fillStyle`
`filRect(x, y, width, height)`
`strokeStyle`
`strokeRect(x, y, width, height)`
`clearRect(x, y, width, height)`



Working with Drawing Objects 2-5

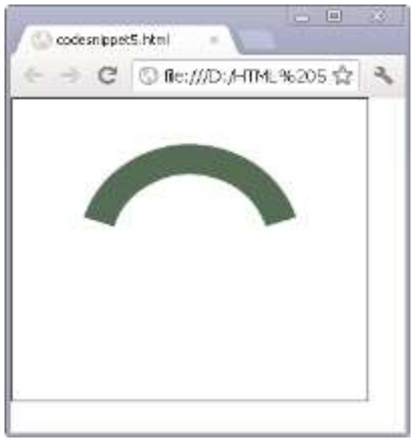
➢ Arcs

- Can create an arc by using the **arc()** method.
- Arcs are represented using a start angle, an end angle, a radius, a center point, and the drawing direction (anticlockwise or clockwise).

Syntax:
arc(x, y, radius, startAngle, endAngle, anticlockwise)

where,

- x, y - the coordinates of the center of an arc
- radius - the distance from the center to any point on the circle
- startAngle, endAngle - the start and end points in the arc
- anticlockwise - Draws the arc clockwise or anticlockwise and accepts a boolean value



Working with Drawing Objects 3-5

➢ Circle

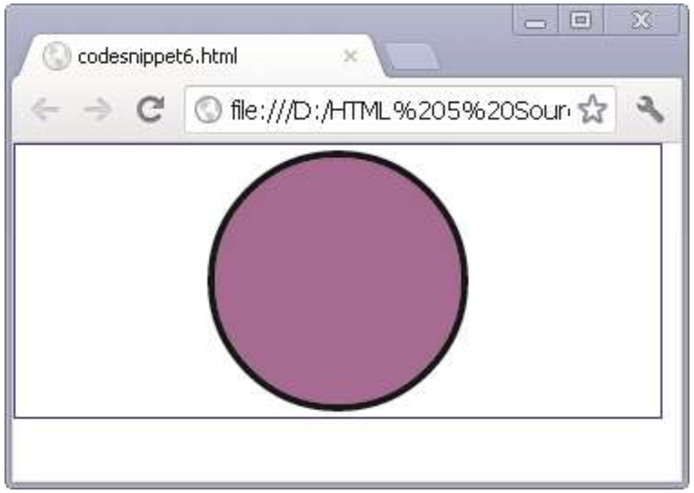
- draw a circle using the **arc()** method.
- have to set the start angle with 0 and the end angle is specified as 2*PI.

Syntax:

```
arc(x, y, radius, startAngle, endAngle, anticlockwise)
```

where,

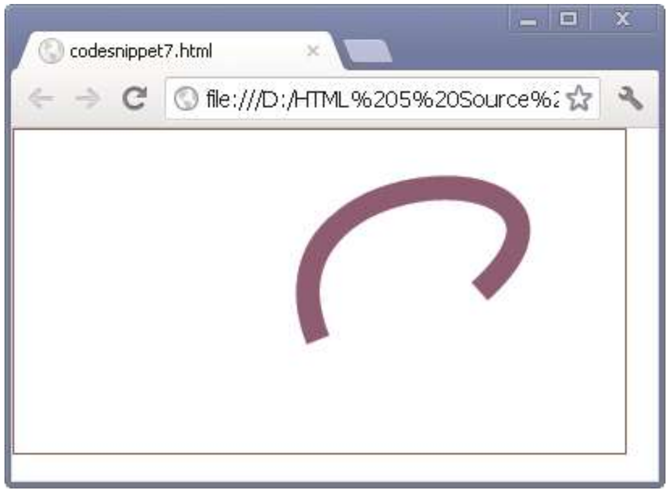
- x, y - Specifies the coordinates of the center of an arc
- radius - Specifies the distance from the center to any point on the circle
- startAngle, endAngle - Specifies the start and end points in the arc
- anticlockwise - Draws the arc clockwise or anticlockwise and accepts a boolean value



Working with Drawing Objects 4-5

➢ Bezier Curves

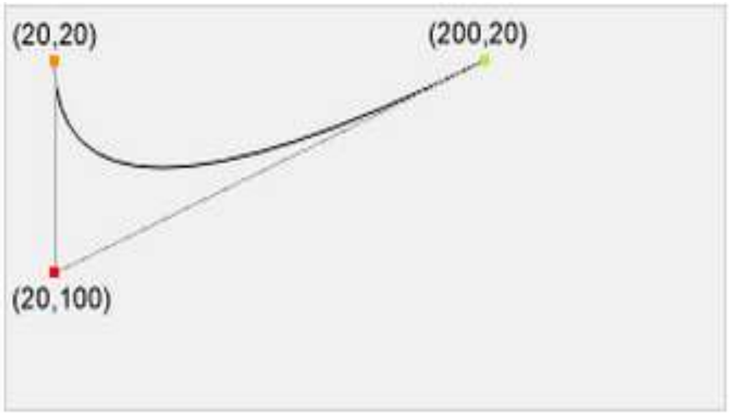
- can create a Bezier curve using the **bezierCurveTo()** method.
- Bezier curves are represented with the two control points, context points, and an end point



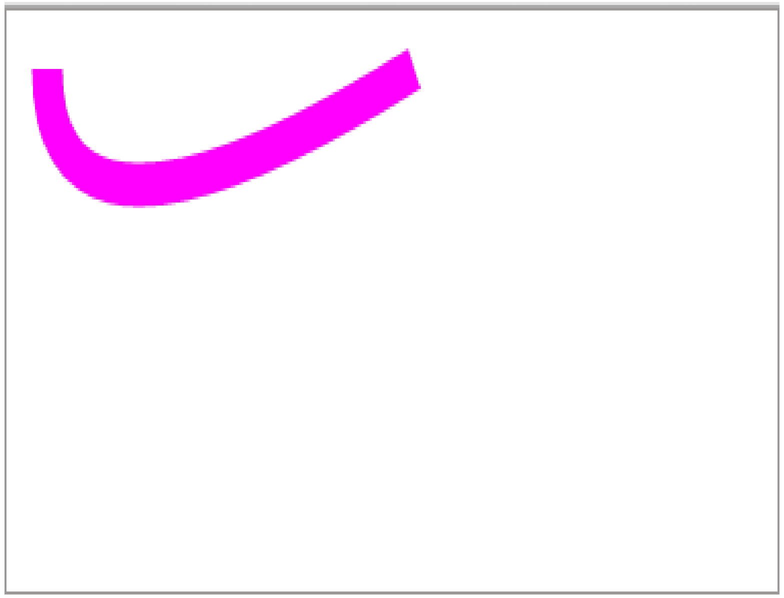
Working with Drawing Objects 5-5

Quadratic Curves

- Can create quadratic curves using the quadraticCurveTo() method.
- Quadratic curves are represented through the context point, an end point, and a control point.

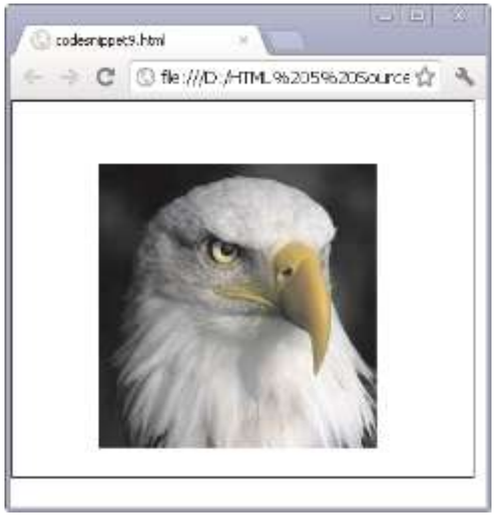


🟧	Start point:	moveTo(**20,20**)
🟥	Control point:	quadraticCurveTo(**20,100**,200,20)
🟩	End point:	quadraticCurveTo(20,100,**200,20**)



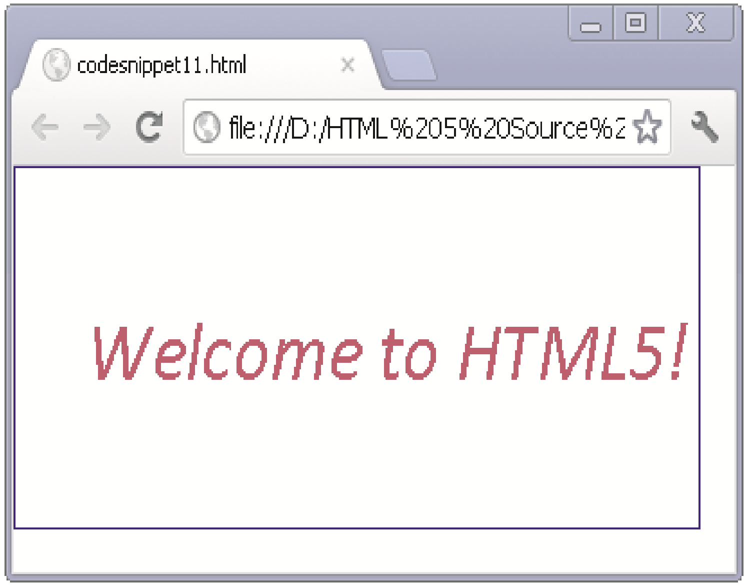
Working with Images

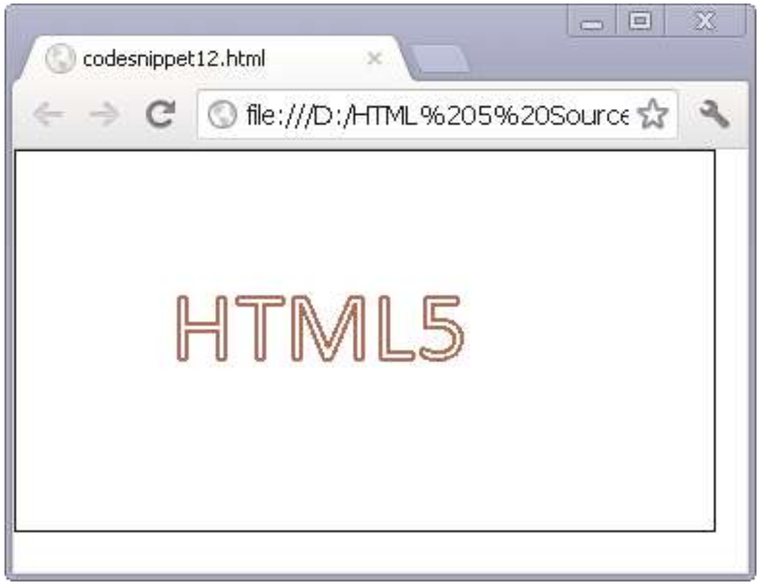
- Image objects can be drawed on canvas using drawImage() method.
- The **drawImage()** method can also draw parts of an image and increase or reduce the size of the image.
- This method accepts nine parameters, depending on editing that is required on the image.
- The image object can be a video, an image, or another canvas element.



Working with Text

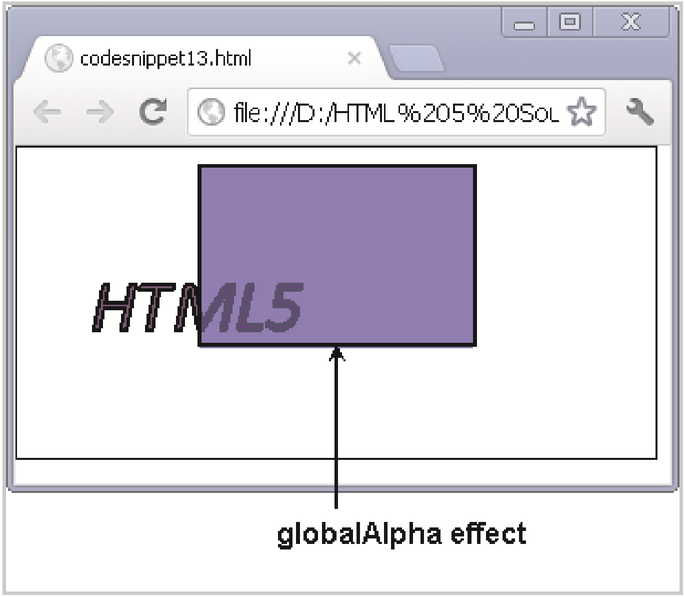
- HTML5 canvas enables you to set the font, style, and size of text by using the font properties.
- The font style can be italic, normal, or bold.
- To set the text color, the fillStyle property of the canvas can be used.





Using Transparency for Text in Canvas

- There are two ways to set the transparency for the text and shapes.
- The first method is to use the strokeStyle and fillStyle by using the rgb function.
- The second method is to use globalAlpha drawing state property, which can be applied universally.
- The globalAlpha property is a value that ranges between 0 (fully transparent) and 1 (fully opaque).

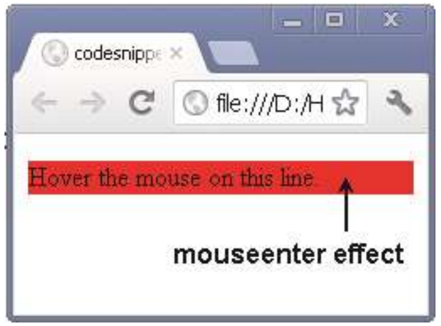


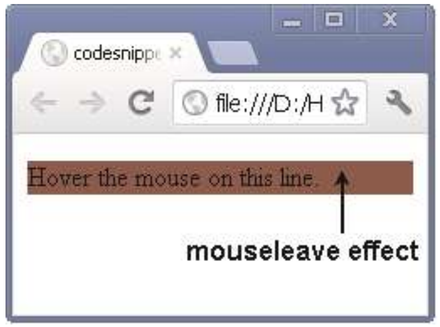
Using Events with jQuery 1-2

- jQuery also offers different events to deal with common interactions when the user moves the mouse or switches between two actions while clicking.
- The following are the events:

➢ hover() event

- The mouseenter and mouseleave are the two events often used together.
- jQuery provides a hover() function that accepts two parameters.
- The first parameter executes when the mouse moves over the element and the second function executes when the mouse moves away from the element.

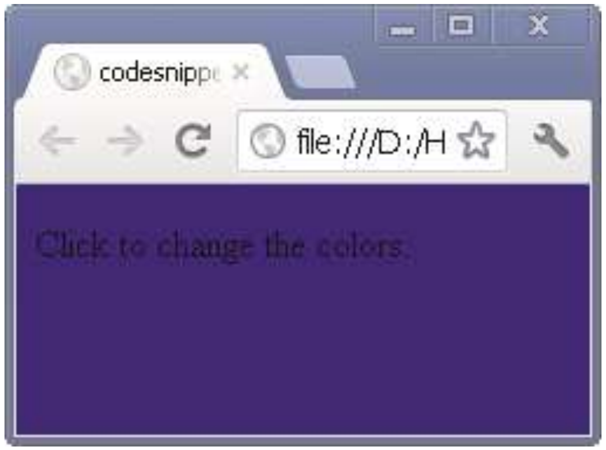




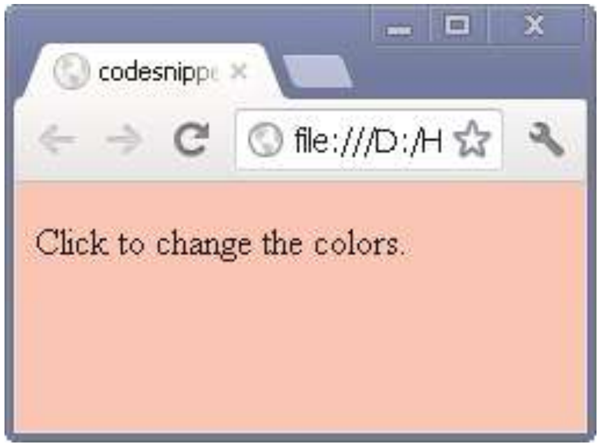
Using Events with jQuery 2-2

➢ toggle() event

- The toggle() event works in a similar manner as that of the hover() event, except that it responds to mouse clicks.
- The toggle() function accepts more than two functions as arguments.
- All the functions passed to the toggle() event will react to its corresponding click action.



Click to change the colors.



Click to change the colors.

Inclusion of External Content in Web Pages

- HTML5 introduces the <eventsource> tag that allows the user to push external content in the Web page. This model is referred to as push model.
- Since the <eventsource> tag is not supported in many browsers, users make use of the <embed> tag for this purpose.
- The <embed> tag is a new element in HTML5 and it is represented as a container for an interactive content or an external application.
- The <embed> tag is often used to add elements such as image, audio, or video on a Web page.
  - The Code Snippet demonstrates the use of <embed> tag.

    ```
    <embed src="mymovie.mp3" />
    ```

  - In this code, the src attribute specifies the path of an external file to embed.

Summary

- The <canvas> element is a drawing area where the user can draw graphics, use images, add animations, and also add text for enhancing the user experience on Web pages.
- To create a line, on a canvas one can use the stroke(), beginPath(), lineTo(), and moveTo() methods.
- Arcs are represented using a start angle, an end angle, a radius, a center point, and the drawing direction (anticlockwise or clockwise).
- With HTML5 canvas, the user can create a rectangle using the rect() method.
- Bezier curves are represented with the two control points, context points, and an end point.
- HTML5 canvas allows the user to create quadratic curves using the quadraticCurveTo() method.
- HTML5 canvas enables the user to draw image object on canvas using the drawImage() method.