



HTML5 API

Objectives

After completing this lesson, you should be able to:

- Exploration of Audio Video API
- Focus Canvas API and SVG
- Drag and Drop API
- Create HTML5 Pages using Geo Location API



Audio Video

Embedding Audio in HTML Document

- Inserting audio onto a web page was not easy before, because web browsers did not have a uniform standard for defining embedded media files like audio.
- Here we'll demonstrate some of the many ways to embed sound in your webpage, from the use of a simple link to the use of the latest HTML5 `<audio>` element.

Using the HTML5 audio Element

- The newly introduced HTML5 <audio> element provides a standard way to embed audio in web pages. However, the audio element is relatively new but it works in most of the modern web browsers.

```
<audio controls="controls" src="media/birds.mp3">  
    Your browser does not support the HTML5 Audio element.  
</audio>
```

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>Embedding Audio into an HTML Page</title>
6  </head>
7  <body>
8      <audio controls="controls" src="Sleep Away.mp3">
9          Your browser does not support the HTML5 audio element.
10     </audio>
11 </body>
12 </html>
```

Using the browser default set of controls, with alternative sources

```
<audio controls="controls">
    <source src="media/birds.mp3" type="audio/mpeg">
    <source src="media/birds.ogg" type="audio/ogg">
        Your browser does not support the HTML5 Audio element.
</audio>
```

- The 'ogg' track in the above example works in Firefox, Opera and Chrome, while the same track in the 'mp3' format is added to make the audio work in Internet Explorer and Safari.

Linking Audio Files

- You can make links to your audio files and play it by ticking on them.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="utf-8">
5      <title>Linking Audio Files in HTML</title>
6  </head>
7  <body>
8      <p><a href="Sleep Away.mp3">Track 1</a></p>
9      <p><a href="Kalimba.mp3">Track 2</a></p>
10 </body>
11 </html>
```

Embedding Video in HTML Document

- Inserting video onto a web page was not relatively easy, because web browsers did not have a uniform standard for defining embedded media files like video.
- Here we'll demonstrates some of the many ways of adding videos on web pages, from the latest HTML5 <video> element to the popular YouTube videos.

Using the HTML5 video Element

- The newly introduced HTML5 <video> element provides a standard way to embed video in web pages. However, the video element is relatively new, but it works in most of the modern web browsers.

```
<video controls="controls" src="media/shuttle.mp4">  
    Your browser does not support the HTML5 Video element.  
</video>
```

Using the embed Element

- The <embed> element is used to embed multimedia content into an HTML document.
- The following code fragment embeds a Flash video into a web page.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  |   <meta charset="utf-8">
5  |   <title>Inserting Video Using embed Element</title>
6  </head>
7  <body>
8  |   <embed src="shuttle.mp4" width="600px" height="200px">
9  </body>
10 </html>
```

Embedding the YouTube Videos

This is the easiest and popular way to embed videos files in the web pages. Just upload the video on YouTube and insert HTML code to display that video in your web page.

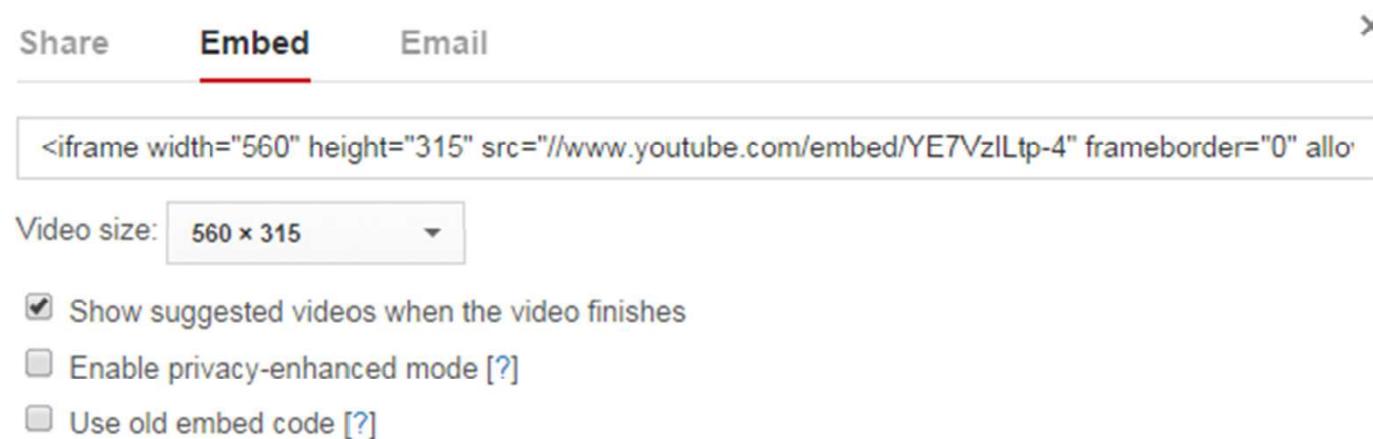
Step 1: Upload video

Go to YouTube upload video page and follow the instructions to upload your video.

Step 2: Creating the HTML Code to embed the video

When you open your uploaded video in YouTube you will see something like the following figure at the bottom of the video. Browse and open your uploaded video in YouTube. Now look for the share button which is located just below the video

- When you click the share button, a share panel will open displaying some more buttons. Now click on the Embed button, it will generate the HTML code to directly embed the video into the web pages. Just copy and paste that code into your HTML document where you want to display the video and you're all set. By default video embedded inside an iframe.



```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title>YouTube Video</title>
</head>
<body>
    <iframe width="560" height="315" src="//www.youtube.com/embed/YE7Vz1Ltp-4" frameborder="0" allowfullscreen></iframe>
</body>
</html>
```



HTML5 Canvas and SVG

HTML5 introduces a canvas element that is used to paint anything by using JavaScript.

```
<canvas width="300" height="200" id="html5canvas">Canvas is not supported.</canvas>
```

If canvas is not supported, the body of the canvas element is rendered.



Drawing Path and Shapes on Canvas

- Here we're going to take a closer look at how to draw basic paths and shapes using the newly introduced HTML5 canvas element and JavaScript.
- Here is the base template for drawing paths and shapes onto the 2D HTML5 canvas.

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4  <meta charset="utf-8">
5  <title>Drawing on Canvas</title>
6  <script>
7      window.onload = function() {
8          var canvas = document.getElementById("myCanvas");
9          var context = canvas.getContext("2d");
10         // draw stuff here
11     };
12 </script>
13 </head>
14 <body>
15     <canvas id="myCanvas" width="300" height="200"></canvas>
16 </body>
17 </html>
```

Painting the Canvas

Use JavaScript to paint the canvas element.

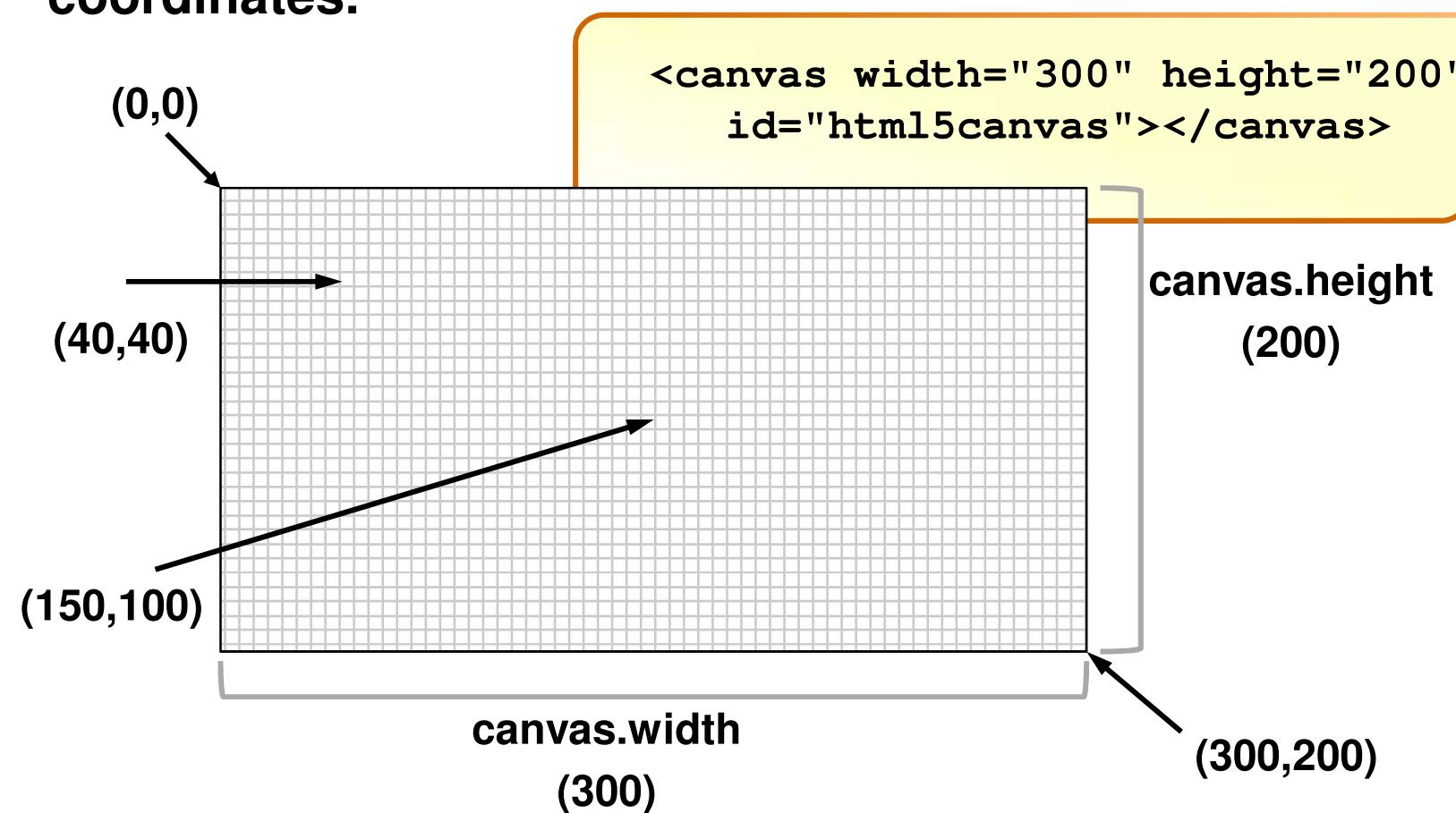
```
var canvasElement = document.getElementById("html5canvas");
// use the context to paint.

var ctx = canvas.getContext("2d");
// Set the fill color with any css3 color.

ctx.fillStyle = "#33DD33"
//fill a rectangle.

ctx.fillRect(x, y, width, height);
```

Most paint functions work on the canvas by providing coordinates.



Canvas Strokes

You can set styles for the stroke and the fill of figures painted on the canvas.

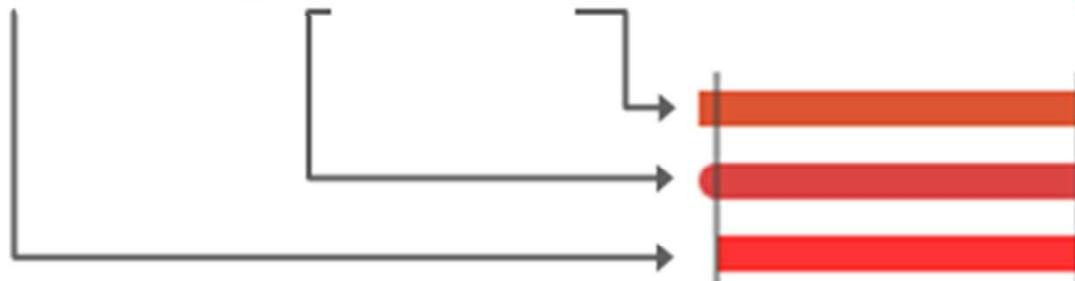
Strokes:

```
ctx.strokeStyle="#000000";
```

Set line width and cap:

```
ctx.lineWidth=3;
```

```
ctx.lineCap="butt" | "round" | "square"
```



Canvas Fills

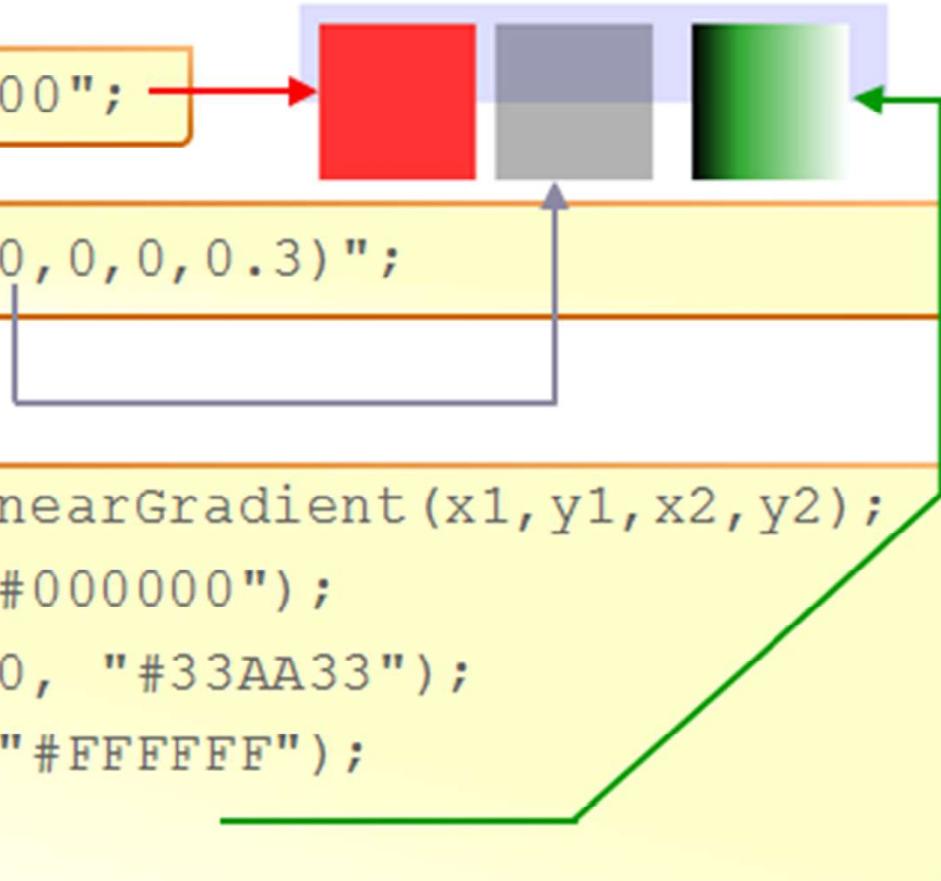
Solid Color:

```
ctx.fillStyle="#FF0000";
```

```
ctx.fillStyle="rgba(0,0,0,0.3);
```

Gradient:

```
var grd=ctx.createLinearGradient(x1,y1,x2,y2);
grd.addColorStop(0,"#000000");
grd.addColorStop(0.30, "#33AA33");
grd.addColorStop(1, "#FFFFFF");
ctx.fillStyle=grd;
```



Canvas Text

You can paint text in a canvas by using:

```
ctx.fillStyle="#000000";
```

```
ctx.font="12px Arial";
ctx.fillText("HELLO!", 10,10);
```



Paths and Strokes

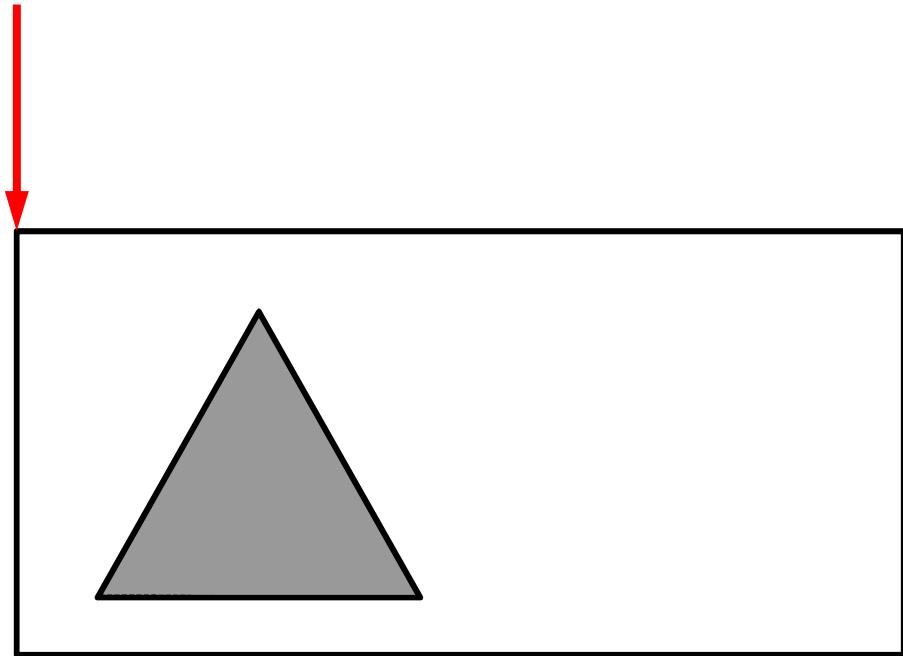
You can create paths to paint complex figures.

```
ctx.beginPath(); //start a path.  
// Path instructions.  
ctx.fill(); // Fill the path.  
ctx.stroke(); //Stroke the path.
```

When using paths, think of how you would move through the shape. Use the “movement” instructions to draw a shape.



Drawing Paths with Lines

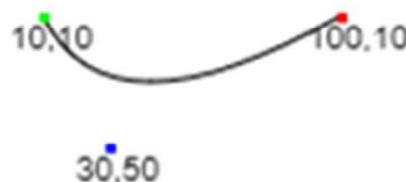


- `ctx.beginPath();`
- `ctx.moveTo(10, 110);`
- `ctx.lineTo(60, 10);`
- `ctx.lineTo(110, 110);`
- `ctx.lineTo(10, 110);`
- `fill();`
- `stroke();`

Path Curves

- `ctx.quadraticCurveTo(cx, cy, x, y);`

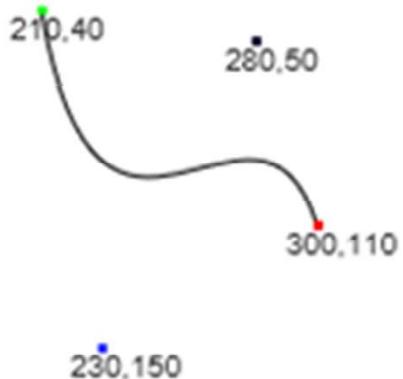
Creates a curve by using one control point



```
ctx.moveTo(10,10);
ctx.quadraticCurveTo(30,50,100,10);
```

- `ctx.bezierCurveTo(c1x, c1y, c2x, c2y, x, y);`

Creates a curve by using two control points

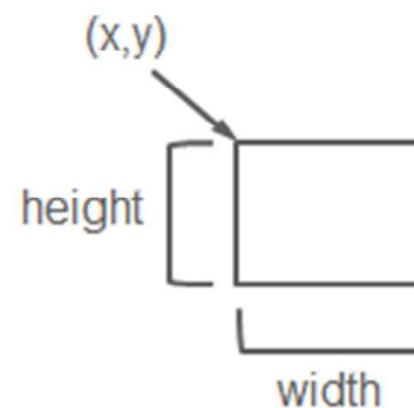


```
ctx.moveTo(210,40);
ctx.bezierCurveTo(230,150,280,50,
300,100);
```

Shapes

Rectangles:

```
ctx.strokeRect(x, y, width, height)  
ctx.fillRect(x, y, width, height)
```



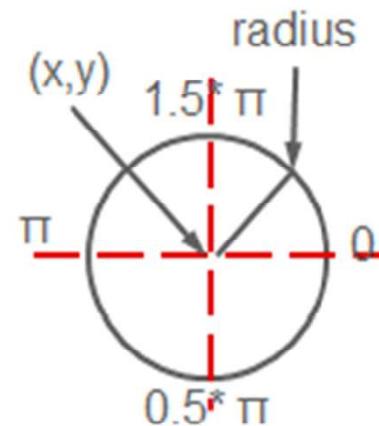
Arcs (circles):

Arcs are paths; you need to begin a path, and then fill or stroke it.

```
ctx.arc(x, y, radius, startAngle, endAngle)
```

To draw a circle, use:

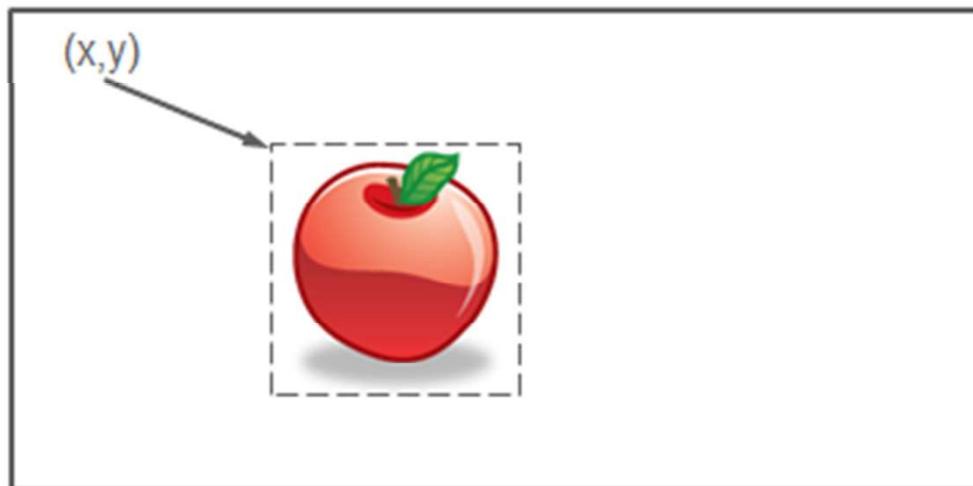
```
ctx.arc(x, y, radius, 0, 2*Math.PI)
```



Painting Images

You can paint an image in a canvas by using an `` element.

```
var img=document.getElementById("imgElement");
ctx.drawImage(img, x, y);
```





HTML5 SVG API

What is SVG?

The Scalable Vector Graphics (SVG) is an XML-based image format that is used to define two-dimensional vector based graphics for the web. Unlike raster image (e.g. .jpg, .gif, .png, etc.), a vector image can be scaled up or down to any extent without losing the image quality.

An SVG image is drawn out using a series of statements that follow the XML schema — that means SVG images can be created and edited with any text editor, such as Notepad. There are several other advantages of using SVG over other image formats like JPEG, GIF, PNG, etc.

- SVG images can be searched, indexed, scripted, and compressed.
- SVG images can be created and modified using JavaScript in real time.
- SVG images can be printed with high quality at any resolution.
- SVG content can be animated using the built-in animation elements.
- SVG images can contain hyperlinks to other documents.

Embedding SVG into HTML Pages

- You can embed SVG graphics directly into your document using the HTML5 `<svg>` element.

```
<body>
  <svg width="300" height="200">
    <text x="10" y="20" style="font-size:14px;">
      Your browser support SVG.
    </text>
    Sorry, your browser does not support SVG.
  </svg>
</body>
```

Drawing a Line

- The most basic path you can draw with SVG is a straight line. The following example will show you how to create a straight line using the SVG <line> element:

```
<style>
  svg {
    border: 1px solid #black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <line x1="50" y1="50" x2="250" y2="150" style="stroke: #red; stroke-width:3;" />
  </svg>
</body>
```

Drawing a Rectangle

- You can create simple rectangle and square shapes using the SVG <rect> element.

```
<head>
  <meta charset="utf-8">
  <title>Create a Rectangle with HTML5 SVG</title>
  <style>
    svg {
      border: 1px solid black;
    }
  </style>
</head>

<body>
  <svg width="300" height="200">
    <rect x="50" y="50"
          width="200" height="100" style="fill: orange; stroke: black; stroke-width:3;" />
  </svg>
</body>
```

Drawing a Circle

- You can also create the circle shapes using the SVG <circle> element.

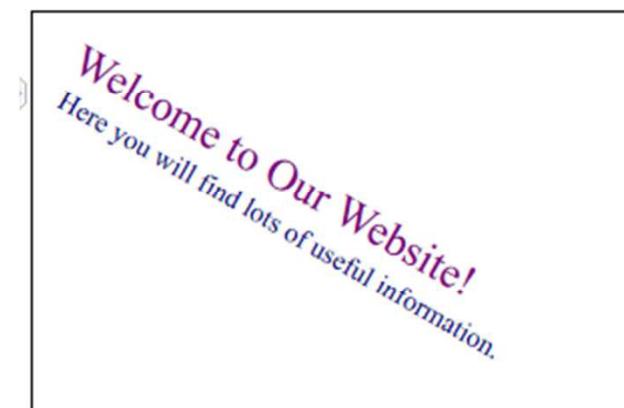
```
6  <style>
7      svg {
8          border: 1px solid #black;
9      }
10 </style>
11 </head>
12 <body>
13     <svg width="300" height="200">
14         <circle cx="150" cy="100" r="70" style="fill:#lime; stroke:#black; stroke-width:3;" />
15     </svg>
16 </body>
```

Drawing Text with SVG

- You can also draw text on the web pages with SVG. The text in SVG is rendered as a graphic so you can apply all the graphic transformation to it but it is still acts like text — that means it can be selected and copied as text by the user.

```
<style>
  svg {
    border: 1px solid black;
  }
</style>
</head>
<body>
  <svg width="300" height="200">
    <text x="20" y="30" style="fill: purple; font-size:22px;">
      Welcome to Our Website!
    </text>
    <text x="20" y="30" dx="0" dy="20" style="fill: navy; font-size:14px;">
      Here you will find lots of useful information.
    </text>
  </svg>
</body>
```

```
1 <svg width="300" height="200">
2   <text x="30" y="15" style="fill:purple; font-size:22px;
transform:rotate(30deg);">
3     <tspan style="fill:purple; font-size:22px;">
4       Welcome to Our Website!
5     </tspan>
6     <tspan dx="-230" dy="20" style="fill:navy; font-size:14px;">
7       Here you will find lots of useful information.
8     </tspan>
9   </text>
10 </svg>
```



Differences between SVG and Canvas

- The HTML5 introduced the two new graphical elements `<canvas>` and `<svg>` for creating rich graphics on the web, but they are fundamentally different.

SVG	Canvas
Vector based (composed of shapes)	Raster based (composed of pixel)
Multiple graphical elements, which become the part of the page's DOM tree	Single element similar to <code></code> in behavior. Canvas diagram can be saved to PNG or JPG format
Modified through script and CSS	Modified through script only
Good text rendering capabilities	Poor text rendering capabilities
Give better performance with smaller number of objects or larger surface, or both	Give better performance with larger number of objects or smaller surface, or both
Better scalability. Can be printed with high quality at any resolution. Pixelation does not occur	Poor scalability. Not suitable for printing on higher resolution. Pixelation may occur



HTML5 Drag and Drop

HTML 5 Drag and Drop

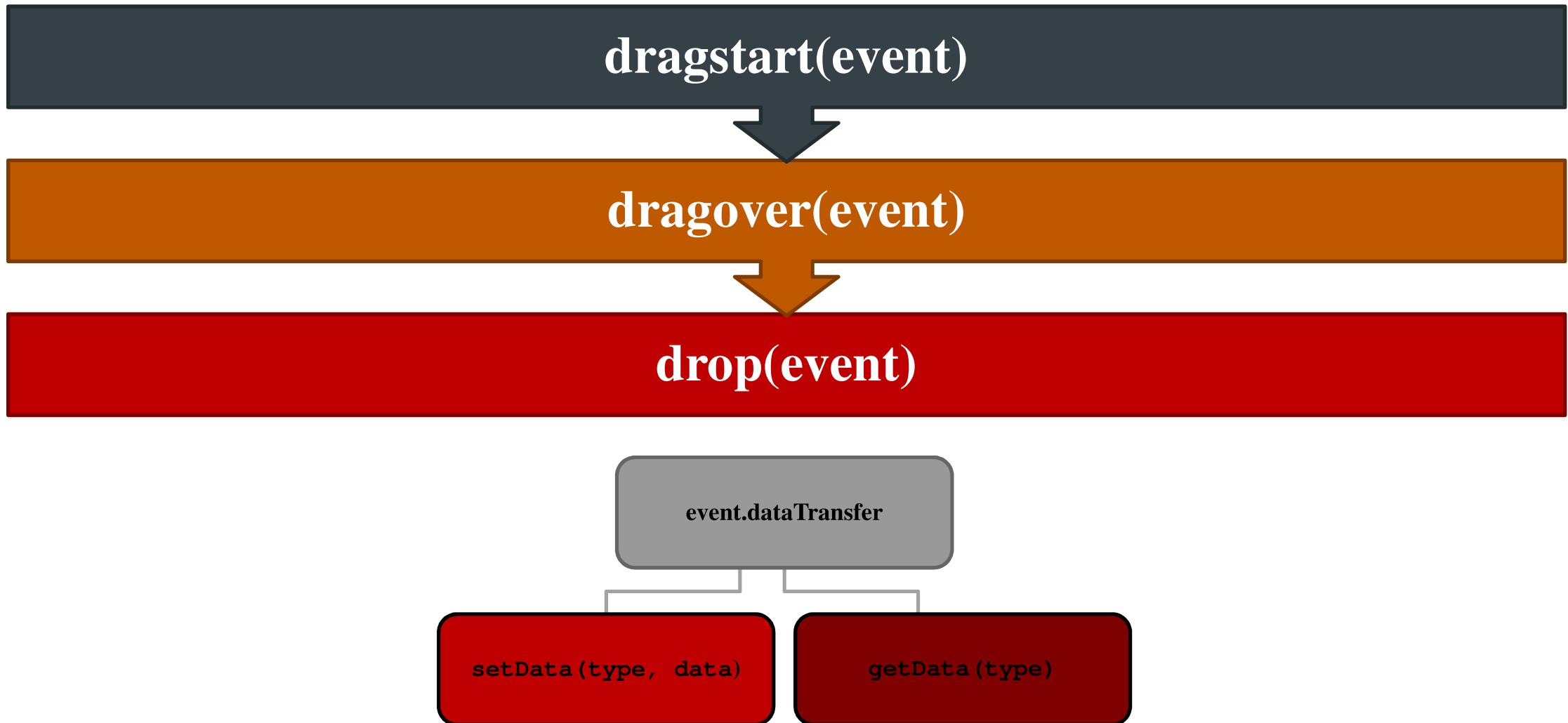
Drag-and-drop interactions are included in HTML5 by using JavaScript events.

In a drag-and-drop operation, usually two elements interact.

The dragged content is stored in a `dataTransfer` object. Using `dataTransfer`, multiple elements can react to drag-and-drop operations.



The Drag-and-Drop Process



Drag Start

The HTML elements that can be dragged to add the following event and set the `draggable` attribute to true include:

```
<div ondragstart="drag(event)"  
draggable="true">Drag Me!</div>
```

The JavaScript code for the handler sets `dataTransfer`:

```
function drag(event) {  
event.dataTransfer.setData("Text", "Dragged");  
}
```

Drag Over

The elements that may receive drags and drops need to add the following event:

```
<div ondragover="dragOver(event)">  
DROP HERE!</div>
```

The JavaScript code is as follows:

```
function dragOver(event) {  
    event.preventDefault();  
}
```

Drop

Elements that receive the drops must add the following listener in addition to onDragOver.

```
<div ondrop="drop(event)"  
ondragover="dragOver(event)"> DROP HERE!</div>
```

The JavaScript code is as follows:

```
function drop(event) {  
    event.preventDefault();  
    alert(event.dataTransfer.getData("Text"));  
}
```



HTML5 Geo Location API

What is Geolocation?

- The HTML5 geolocation feature lets you find out the geographic coordinates (latitude and longitude numbers) of the current location of your website's visitor.
- This feature is helpful for providing better browsing experience to the site visitor. For example, you can return the search results that are physically close to the user's location.

Finding a Visitor's Coordinates

- Getting the position information of the site visitor using the HTML5 geolocation API is fairly simple. It utilizes the three methods that are packed into the navigator.geolocation object — getCurrentPosition(), watchPosition() and clearWatch().
- The simple example of geolocation that displays your current position. But, first you need to agree to let the browser tell the web server about your position.

Showing Location on Google Map

- You can do very interesting things with geolocation data, like showing the user location on Google map. The following example will show your current location on Google map based the latitude and longitude data retrieved through the HTML5 geolocation feature.

In this lesson, you should have learned how to:

- Exploration of Audio Video API
- Focus Canvas API and SVG
- Drag and Drop API
- Create HTML5 Pages using Geo Location API

