What is HTML Canvas?

The HTML <canvas> element is used to draw graphics, on the fly, via scripting (usually JavaScript).

The <canvas> element is only a container for graphics. You must use a script to actually draw the graphics.

Canvas has several methods for drawing paths, boxes, circles, text, and adding images.

HTML Canvas Can Draw Text

Canvas can draw colorful text, with or without animation.

HTML Canvas Can Draw Graphics

Canvas has great features for graphical data presentation with an imagery of graphs and charts.

HTML Canvas Can be Animated

Canvas objects can move. Everything is possible: from simple bouncing balls to complex animations.

HTML Canvas Can be Interactive

Canvas can respond to JavaScript events.

Canvas can respond to any user action (key clicks, mouse clicks, button clicks, finger movement).

HTML Canvas Can be Used in Games

Canvas' methods for animations, offer a lot of possibilities for HTML gaming applications.

Canvas Example

In HTML, a <canvas> element looks like this:

<canvas id="myCanvas" width="200" height="100"></canvas>

The <canvas> element must have an id attribute so it can be referred to by JavaScript.

The width and height attribute is necessary to define the size of the canvas.

**Tip:** You can have multiple <canvas> elements on one HTML page.

|  |  |
| --- | --- |
| **ote** | By default, the <canvas> element has no border and no content. |

To add a border, use a style attribute:

Example

<canvas id="myCanvas" width="200" height="100"  
style="border:1px solid #000000;">  
</canvas>

|  |  |
| --- | --- |
| **ote** | **Tip:** You can have multiple <canvas> elements on one HTML page. |

The next chapters show how to draw on the canvas.

Draw on the Canvas With JavaScript

All drawing on the HTML canvas must be done with JavaScript:

Example

<script>  
var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.fillStyle = "#FF0000";  
ctx.fillRect(0,0,150,75);  
</script>

Step 1: Find the Canvas Element

First of all, you must find the <canvas> element.

This is done by using the HTML DOM method getElementById():

var canvas = document.getElementById("myCanvas");

Step 2: Create a Drawing Object

Secondly, you need a drawing object for the canvas.

The getContext() is a built-in HTML object, with properties and methods for drawing:

var ctx = canvas.getContext("2d");

Step 3: Draw on the Canvas

Finally, you can draw on the canvas.

Set the fill style of the drawing object to the color red:

ctx.fillStyle = "#FF0000";

The fillStyle property can be a CSS color, a gradient, or a pattern. The default fillStyle is black.

The fillRect(*x,y,width,height*) method draws a rectangle, filled with the fill style, on the canvas:

ctx.fillRect(0,0,150,75);

Canvas Coordinates

The HTML canvas is a two-dimensional grid.

The upper-left corner of the canvas has the coordinates (0,0)

In the previous chapter, you saw this method used: fillRect(0,0,150,75).

This means: Start at the upper-left corner (0,0) and draw a 150x75 pixels rectangle.

Coordinates Example

Mouse over the rectangle below to see its x and y coordinates:

X

Y

Draw a Line

To draw a straight line on a canvas, use the following methods:

* moveTo(*x,y*) - defines the starting point of the line
* lineTo(*x,y*) - defines the ending point of the line

To actually draw the line, you must use one of the "ink" methods, like stroke().

Example

Define a starting point in position (0,0), and an ending point in position (200,100). Then use the stroke() method to actually draw the line:

var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.moveTo(0,0);  
ctx.lineTo(200,100);  
ctx.stroke();

Draw a Circle

To draw a circle on a canvas, use the following methods:

* beginPath();
* arc(x,y,r,start,stop)

Example

Define a circle with the arc() method. Then use the stroke() method to actually draw the circle:

var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.beginPath();  
ctx.arc(95,50,40,0,2\*Math.PI);  
ctx.stroke();

Canvas - Gradients

Gradients can be used to fill rectangles, circles, lines, text, etc. Shapes on the canvas are not limited to solid colors.

There are two different types of gradients:

* createLinearGradient(*x,y,x1,y1*) - creates a linear gradient
* createRadialGradient(*x,y,r,x1,y1,r1*) - creates a radial/circular gradient

Once we have a gradient object, we must add two or more color stops.

The addColorStop() method specifies the color stops, and its position along the gradient. Gradient positions can be anywhere between 0 to 1.

To use the gradient, set the fillStyle or strokeStyle property to the gradient, then draw the shape (rectangle, text, or a line).

Using createLinearGradient()

Example

Create a linear gradient. Fill rectangle with the gradient:

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
  
// Create gradient  
var grd=ctx.createLinearGradient(0,0,200,0);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle=grd;  
ctx.fillRect(10,10,150,80);

Using createRadialGradient():

Example

Create a radial/circular gradient. Fill rectangle with the gradient:

JavaScript:

var c=document.getElementById("myCanvas");  
var ctx=c.getContext("2d");  
  
// Create gradient  
var grd=ctx.createRadialGradient(75,50,5,90,60,100);  
grd.addColorStop(0,"red");  
grd.addColorStop(1,"white");  
  
// Fill with gradient  
ctx.fillStyle = grd;  
ctx.fillRect(10,10,150,80);

Drawing Text on the Canvas

To draw text on a canvas, the most important property and methods are:

* font - defines the font properties for the text
* fillText(*text,x,y*) - draws "filled" text on the canvas
* strokeText(*text,x,y*) - draws text on the canvas (no fill)

Using fillText()

Example

Set font to 30px "Arial" and write a filled text on the canvas:

JavaScript:

var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.font = "30px Arial";  
ctx.fillText("Hello World",10,50);

Using strokeText()

Example

Set font to 30px "Arial" and write a text, with no fill, on the canvas:

JavaScript:

var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.font = "30px Arial";  
ctx.strokeText("Hello World",10,50);

Add Color and Center Text

Example

Set font to 30px "Comic Sans MS" and write a filled red text in the center of the canvas:

JavaScript:

var canvas = document.getElementById("myCanvas");  
var ctx = canvas.getContext("2d");  
ctx.font = "30px Comic Sans MS";  
ctx.fillStyle = "red";  
ctx.textAlign = "center";  
ctx.fillText("Hello World", canvas.width/2, canvas.height/2);