

# HTML5: Canvas API

1. [浏览器支持](#)
2. [API 参考](#)
3. [DEMO](#)
4. [注意事项](#)
5. [兼容解决方案](#)
6. [参考文献](#)

## 一、浏览器支持

Platform	MAC						
Browser	<a href="#">OPERA</a>	<a href="#">FIREFOX</a>	<a href="#">SAFARI</a>	<a href="#">CHROME</a>	<a href="#">OPERA</a>	<a href="#">FIREFOX</a>	
Version	10.63	3.6	5	7	10.63	3.6	4.03
Canvas	√	√	√	√	√	√	√
Canvas Text	√	√	√	√	√	√	√

## 二、API 参考

1. **API SPEC:** <http://www.whatwg.org/specs/web-apps/current-work/multipage/the-canvas-element.html>

- `<canvas />`

### 2. HTML 写法 :

Canvas 生成的时候默认是不可见的。但我们可以通过 CSS 来改变它的外观。其 HTML 写法也像其他 HTML5 新增的标签一样简洁。Global Attribute 也只有 height 和 width , 主要还是用 Javascript 来对其进行操作。

```
<canvas id="diagonal" style="border: 1px solid;" width="200"
height="200">
    canvas is not supported
</canvas>
```

### 3. Script

```
function drawDiagonal() {

    // Get the canvas element and its drawing context
    var canvas = document.getElementById('diagonal');
    var context = canvas.getContext('2d');

    // Create a path in absolute coordinates
    context.beginPath();
    context.moveTo(70, 140);
    context.lineTo(140, 70);

    // Stroke the line onto the canvas
    context.stroke();
}

window.addEventListener("load", drawDiagonal, true);
```

### (1) 浏览器检测

```
try {
    var hasCanvas =
        !! (document.createElement('canvas').getContext('2d'));
} catch(msg) {}
```

### (2) Scripting API

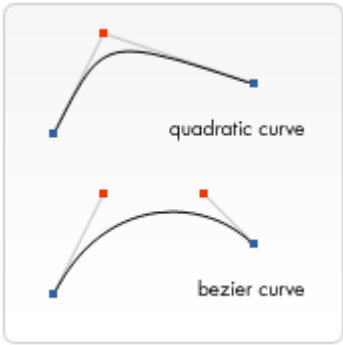
- Canvas: var canvas = document.getElementById('canvas')

canvas 对象属性	描述
toDataURL([Optional], [Variadic])	返回 string
getContext(string Context Type)	返回 object

- Context: var ctx = canvas.getContext('2d');

注：x, y 通常代指坐标点的 x 和 y轴

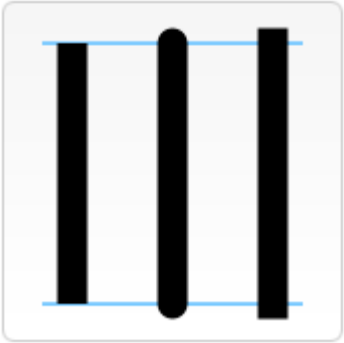

矩形	描述
fillRect(x, y, width, height)	画一个矩形
strokeRect(x, y, width, height)	画一个矩形外框
clearRect(x, y, width, height)	类似于打孔效果，把某个层清除，将其变成透明区
rect(x, y, width, height)	x,y 新矩形的 top 和 left 值。当这个方法被调用，默认调用 moveTo(0,0)

绘制路径	描述
<code>beginPath()</code>	告诉浏览器，你要开始绘制。当 <code>beginPath()</code> 被调用，是另一相当于 <code>html</code> 中的起始标签。
<code>closePath()</code>	结构一个绘制过程。 相当于 <code>html</code> 中的结束标签。
<code>stroke()</code>	绘制形状外框
<code>fill()</code>	绘制实心形状。一旦 <code>fill()</code> 被调用，形状的绘制将被结束，不
<code>clip()</code>	把后续特性限制在剪接的图形内。
<code>moveTo(x, y)</code>	描述虚拟的“画笔”放到要开始绘制的点
<code>lineTo(x, y)</code>	绘制线条， <code>x</code> ， <code>y</code> 表示线条的结束点坐标。
<code>arc(x, y, radius, startAngle, endAngle, anticlockwise)</code>	绘制弧形、圆形。 <code>x</code> ， <code>y</code> 是圆的中心坐标； <code>radius</code> 半径； <code>startAngle</code> 弧度的开始点和结束点，以 <code>x</code> 轴为准； <code>anticlockwise</code> ，逆时针旋  注： <code>Angle</code> 是靠弧度来计算的，而不是以旋转的度数来计算。 <code>var degree = 270, // 度数</code> <code>radians = (Math.PI/180)*degree; // 弧度</code>
<code>quadraticCurveTo(cplx, cply, x, y)</code>	<code>x</code> ， <code>y</code> 为结束点坐标； <code>cplx</code> ， <code>cply</code> 为第一个控制点的坐标；同上控制点的坐标。  注： <code>quadratic curve</code> 只有一个控制点，而 <code>bezier curve</code> 有 1 写成字母 <code>L</code> 的小写。
<code>bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y)</code>	

图片	描述
<code>drawImage(image, x, y [, width, height])</code>	1. <code>image</code> ，图片，或者我们的 <code>canvas</code> 对象。如 <code>var img = new image();</code>

	<div>2. x, y 在 canvas 中放置图片的坐标，想当于 CSS 的 top, left;</div> <div>3. width, height ,生成图片的宽高。</div>
<div>drawImage(image, sx, sy, sWidth, sHeight, dx, dy, dWidth, dHeight)</div>	<div>1. image 同上 ;</div> <div>2. 其他属性用图示比较容易理解 :</div> <div>s = source, d = destination canvas ;</div> <div>sx, sy, 截取原始变成目标图片的开始坐标点 ;</div> <div>swidth, sheight 想实现目标图片的大小 ;</div> <div>dx, dy, 目标图片在画面中的位置坐标点 ;</div> <div>dwidth, dheight 目标图片在画布中的大小。</div> <div></div>
<div>createPattern(image, type)</div>	<div>1. image 同上 ;</div> <div>2. type 有 repeat, repeat-x, repeat-y, no-repeat。意思同 CSS 中的 background-repeat。</div>

颜色	描述
<div>fillStyle = color</div>	<div>填充颜色</div> <div>注 : color 可以是 CSS 颜色中的任何一种。<a href="#">CSS3 color values</a></div>
<div>strokeStyle = color</div>	<div>绘制线条颜色。color value 同上</div>
<div>globalAlpha = transparency value</div>	<div>transparency value 可以在 0.0 ~ 1.0 之间的值。表示 alpha 透明。类似于设置 CSS 的 opacity。</div>

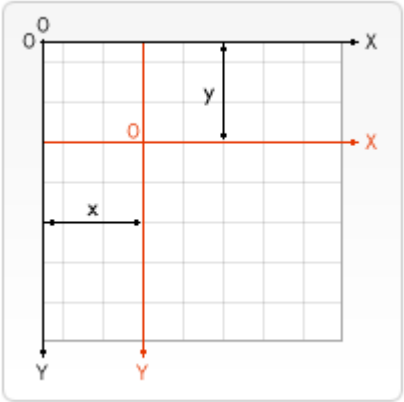
线条	描述
<code>lineWidth = value</code>	线条的大小。必须是正值。默认值是 1.0。
<code>lineCap = type</code>	<p>线条起始点模式。type 有 3 种：butt/round/square。默认为 butt。</p> <p>注：butt 与 square 的区别是，square 的半个 square 在基准线给的两端突出。如图：</p> 
<code>lineJoin = type</code>	<p>线组合模式。type 有 3 种：round/bevel/miter。默认为 miter。</p> 
<code>miterLimit = value</code>	线条组件斜面的最大值。

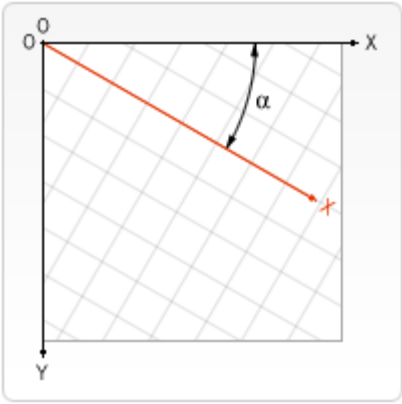
渐变	描述
<code>createLinearGradient(x1, y1, x2, y2)</code>	线性渐变。x1, y1, 渐变起始点的坐标；x2, y2 渐变结束点的坐标轴。
<code>createRadialGradient(x1, y1, r1, x2, y2, r2)</code>	放射线渐变。x1, y1, x2, y2 同上。r1 表示起始点的半径，r2 表示结束点的半径。
<code>addColorStop(position, color)</code>	添加渐变起始点或结束点的渐变颜色。position 的值是：0.0~1.0；Color 为 CSS3 允许的颜色值。

	<p>注：用于 gradient 上。</p> <pre>var lGradient = ctx.createLinearGradient(0, 0, 0, 150); lGradient.addColorStop(0.6, 'red');</pre>
--	--

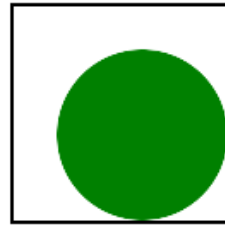
投影	描述
shadowOffsetX = value	x 方面投影大小，{Number}
shadowOffsetY = value	y 方向投影大小，{Number}
shadowBlur = value	扩散投影大小，{Number}
shadowColor = color	投影颜色。color, CSS3 支持的颜色

存储	描述
save()	保存。
restore()	恢复。一旦被调用，将会恢复到上一次保存的样式。

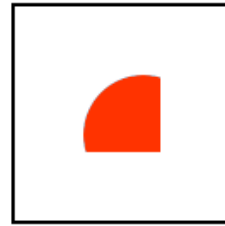
变形	描述
<p>translate(x, y)</p>	<p>把 canvas 的原点行动到 x, y 作为其新原点。</p> 
<p>rotate(angle)</p>	<p>顺时间旋转。angle 是指弧度，像 arc 的 startAngle 和 endAngle。</p>

	
<code>scale(x, y)</code>	<code>x</code> , <code>y</code> 水平和垂直方向的缩放比率，值在 $-1 \sim 1$ 之间。
<code>transform(m11, m12, m21, m22, dx, dy)</code>	变换矩阵 (transformation matrix)。官方 和 MDC 的描述都不是很清楚。大家用到的时候搜索一下。
<code>setTransform(m11, m12, m21, m22, dx, dy)</code>	设置变形的值。参数同上。

组合	描述
<code>globalCompositeOperation = type</code>	共 12 种状态。有多层的状态下。会稍有不同。大家自己手写一下，印点。分别如下图显示：



source-over



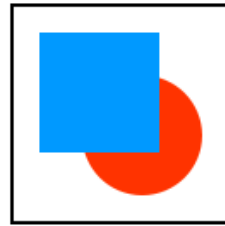
source-in



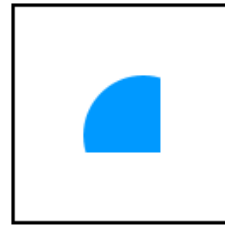
source-out



source-atop



destination-over



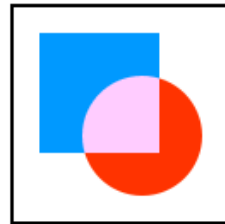
destination-in



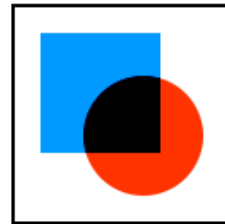
destination-out



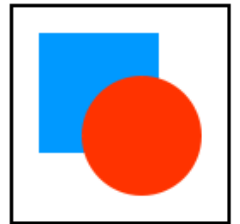
destination-atop



lighter



darker














copy



xor

#### 4. Cheetsheet



Canvas element			Compositing			Colors, styles and shadows			Text		
Attributes			Attributes			Attributes			Attributes		
Name	Type	Default	Name	Type	Default	Name	Type	Default	Name	Type	Default
width	unsigned long	300	globalAlpha	float	1.0	strokeStyle	any	black	font	string	10px sans-serif
height	unsigned long	150	globalCompositeOperation	string	source-over	fillStyle	any	black	textAlign	string	start
Methods			Supports any of the following values:			shadowOffsetX	float	0.0	Supports any of the following values: [start, end, left, right, center]		
Return	Name					shadowOffsetY	float	0.0	textBaseline	string	alphabetic
string	toDataURL( [Optional] string type, [Variadic] any args)					shadowBlur	float	0.0	Supports any of the following values: [top, hanging, middle, alphabetic, ideographic, bottom]		
Object	getContext(string extid)					shadowColor	string	transparent black	Methods		
2D Context						Methods			Return	Name	
Attributes			Line styles			CanvasGradient	createLinearGradient( float x0, float y0, float x1, float y1)		void	fillText( string text, float x, float y, [Optional] float maxWidth)	
Name	Type		Name	Type	Default	CanvasGradient	createRadialGradient( float x0, float y0, float r0, float x1, float y1, float r1)		void	strokeText( string text, float x, float y, [Optional] float maxWidth)	
canvas	HTMLCanvasObject [readonly]		lineWidth	float	1.0	CanvasPattern	createPattern( Object image, string repetition)		void	measureText(string text)	
Methods			lineCap	string	butt	Argument "image" can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement			TextMetrics interface		
Return	Name		Supports any of the following values:			"repetition" supports any of the following values: [repeat (default), repeat-x, repeat-y, no-repeat]			width	float	[readonly]
void	save()		butt	round	square	CanvasGradient interface			Rectangles		
void	restore()		miter	bevel	miter	void	addColorStop( float offset, string color)		Methods		
Transformation			Supports any of the following values:			CanvasPattern interface			Return	Name	
Return	Name		miterLimit	float	10	No attributes or methods.	Paths		void	clearRect( float x, float y, float w, float h)	
void	scale(float x, float y)		Image drawing			void	beginPath()		void	fillRect( float x, float y, float w, float h)	
void	rotate(float angle)		Methods			void	closePath()		void	fillRect( float x, float y, float w, float h)	
void	translate(float x, float y)		Return	Name		void	stroke()		void	strokeRect( float x, float y, float w, float h)	
void	transform( float m11, float m12, float m21, float m22, float dx, float dy)		void	drawImage( Object image, float dx, float dy, [Optional] float dw, float dh)		void	clip()		void	strokeRect( float x, float y, float w, float h)	
void	setTransform( float m11, float m12, float m21, float m22, float dx, float dy)		void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	moveTo(float x, float y)		Pixel manipulation		
Image drawing			Argument "image" can be of type HTMLImageElement, HTMLCanvasElement or HTMLVideoElement			void	lineTo(float x, float y)		Methods		
Return	Name		void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	quadraticCurveTo( float cp1x, float cp1y, float cp2x, float cp2y, float x, float y)		Return	Name	
void	drawImage( Object image, float dx, float dy, [Optional] float dw, float dh)		void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	bezierCurveTo( float cp1x, float cp1y, float cp2x, float cp2y, float x, float y)		ImageData	createImageData(float sw, float sh)	
void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	arcTo( float x1, float y1, float x2, float y2, float radius)		ImageData	createImageData(imageData)	
Methods			void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	arc( float x, float y, float radius, float startAngle, float endAngle, boolean anticlockwise)		ImageData	getImageData( float sx, float sy, float sw, float sh)	
void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	rect(float x, float y, float w, float h)		void	putImageData(imageData, float dx, float dy, [Optional] float dirtyX, float dirtyY, float dirtyWidth, float dirtyHeight)	
ImageData interface			void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void	isPointInPath(float x, float y)		ImageData interface		
width	unsigned long	[readonly]	void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		boolean			width	unsigned long	[readonly]
height	unsigned long	[readonly]	void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void			height	unsigned long	[readonly]
data	CanvasPixelArray	[readonly]	void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void			data	CanvasPixelArray	[readonly]
CanvasPixelArray interface			void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void			CanvasPixelArray interface		
length	unsigned long	[readonly]	void	drawImage( Object image, float sx, float sy, float sw, float sh, float dx, float dy, float dw, float dh)		void			length	unsigned long	[readonly]

Source: <http://www.whatwg.org/specs/web-apps/current-work/> (2009-05-04)

### 三、DEMO

[Deviant Art Muro](#)

### 四、注意事项

- 创建的 Canvas 默认是不可见的，需要用 CSS 来样式化。
- 如果不设置宽度，默认宽高分别是 300px 150px。
- 目前 Context Type 只有 2D 是被支持的，因此检测其支持状态时，我们用的是 `getContext('2D')`。
- 外观都是根据其 Context 来创建的，而非 Canvas 本身。
- Canvas 绘制的是位图，而不是 SVG 那种矢量图。

### 五、兼容解决方案

目前比较好的兼容方案是，在 IE6-8 中使用开源的 Explore Canvas。

Open source on Google Code: <http://code.google.com/p/explorercanvas/>

## 六、参考文献

- BOOK: 《Pro HTML5 Programming》
- [Canvas tutorial](#)
- [HTML5 Spec: the canvas element](#)
- [HTML5 Canvas CheetSheet](#)
- [HTML5 & CSS3 Support, Web Design Tools & Support](#)