Где почитать про SVG:

<https://www.w3schools.com/graphics/game_canvas.asp>

Examples in Each Chapter

With our "Try it Yourself" editor, you can edit the SVG, and click on a button to view the result.

### **SVG Example**

<html>  
<body>  
  
<h1>My first SVG</h1>

<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>  
  
</body>  
</html>

## What you should already know

Before you continue, you should have some basic understanding of the following:

* HTML
* Basic XML

If you want to study these subjects first, find the tutorials on our [Home page](https://www.w3schools.com/default.asp).

## What is SVG?

* SVG stands for Scalable Vector Graphics
* SVG is used to define vector-based graphics for the Web
* SVG defines the graphics in XML format
* Every element and every attribute in SVG files can be animated
* SVG is a W3C recommendation
* SVG integrates with other W3C standards such as the DOM and XSL

## SVG is a W3C Recommendation

SVG 1.0 became a W3C Recommendation on 4 September 2001.

SVG 1.1 became a W3C Recommendation on 14 January 2003.

SVG 1.1 (Second Edition) became a W3C Recommendation on 16 August 2011.

## SVG Advantages

Advantages of using SVG over other image formats (like JPEG and GIF) are:

* SVG images can be created and edited with any text editor
* SVG images can be searched, indexed, scripted, and compressed
* SVG images are scalable
* SVG images can be printed with high quality at any resolution
* SVG images are zoomable
* SVG graphics do NOT lose any quality if they are zoomed or resized
* SVG is an open standard
* SVG files are pure XML

The main competitor to SVG is Flash.

The biggest advantage SVG has over Flash is the compliance with other standards (e.g. XSL and the DOM). Flash relies on proprietary technology that is not open source.

## Creating SVG Images

SVG images can be created with any text editor, but it is often more convenient to create SVG images with a drawing program, like [Inkscape](http://inkscape.org/" \t "_blank).

# **SVG in HTML**

In HTML5, you can embed SVG elements directly into your HTML pages.

## Embed SVG Directly Into HTML Pages

Here is an example of a simple SVG graphic:

and here is the HTML code:

### **Example**

<!DOCTYPE html>  
<html>  
<body>  
  
<h1>My first SVG</h1>

<svg width="100" height="100">  
  <circle cx="50" cy="50" r="40" stroke="green" stroke-width="4" fill="yellow" />  
</svg>  
  
</body>  
</html>

**SVG Code explanation:**

* An SVG image begins with an <svg> element
* The width and height attributes of the <svg> element define the width and height of the SVG image
* The <circle> element is used to draw a circle
* The cx and cy attributes define the x and y coordinates of the center of the circle. If cx and cy are not set, the circle's center is set to (0, 0)
* The r attribute defines the radius of the circle
* The stroke and stroke-width attributes control how the outline of a shape appears. We set the outline of the circle to a 4px green "border"
* The fill attribute refers to the color inside the circle. We set the fill color to yellow
* The closing </svg> tag closes the SVG image

**Note:** Since SVG is written in XML, all elements must be properly closed!

# **SVG <rect>**

SVG Shapes

SVG has some predefined shape elements that can be used by developers:

* Rectangle <rect>
* Circle <circle>
* Ellipse <ellipse>
* Line <line>
* Polyline <polyline>
* Polygon <polygon>
* Path <path>

The following chapters will explain each element, starting with the rect element.

## SVG Rectangle - <rect>

## Example 1

The <rect> element is used to create a rectangle and variations of a rectangle shape:

Here is the SVG code:

### **Example**

<svg width="400" height="110">  
  <rect width="300" height="100" style="fill:rgb(0,0,255);stroke-width:3;stroke:rgb(0,0,0)" />  
</svg>

**Code explanation:**

* The width and height attributes of the <rect> element define the height and the width of the rectangle
* The style attribute is used to define CSS properties for the rectangle
* The CSS fill property defines the fill color of the rectangle
* The CSS stroke-width property defines the width of the border of the rectangle
* The CSS stroke property defines the color of the border of the rectangle

## Example 2

Let's look at another example that contains some new attributes:

Here is the SVG code:

### **Example**

<svg width="400" height="180">  
  <rect x="50" y="20" width="150" height="150"  
  style="fill:blue;stroke:pink;stroke-width:5;fill-opacity:0.1;stroke-opacity:0.9" />  
</svg>

**Code explanation:**

* The x attribute defines the left position of the rectangle (e.g. x="50" places the rectangle 50 px from the left margin)
* The y attribute defines the top position of the rectangle (e.g. y="20" places the rectangle 20 px from the top margin)
* The CSS fill-opacity property defines the opacity of the fill color (legal range: 0 to 1)
* The CSS stroke-opacity property defines the opacity of the stroke color (legal range: 0 to 1)

## Example 3

Define the opacity for the whole element:

Here is the SVG code:

### **Example**

<svg width="400" height="180">  
  <rect x="50" y="20" width="150" height="150"  
  style="fill:blue;stroke:pink;stroke-width:5;opacity:0.5" />  
</svg>

**Code explanation:**

* The CSS opacity property defines the opacity value for the whole element (legal range: 0 to 1)

Example 4

Last example, create a rectangle with rounded corners:

Here is the SVG code:

### **Example**

<svg width="400" height="180">  
  <rect x="50" y="20" rx="20" ry="20" width="150" height="150"  
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />  
</svg>

**Code explanation:**

* The rx and the ry attributes rounds the corners of the rectangle