**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 omitted, 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!

The <path> element is used to define a path.

The following commands are available for path data:

* M = moveto
* L = lineto
* H = horizontal lineto
* V = vertical lineto
* C = curveto
* S = smooth curveto
* Q = quadratic Bézier curve
* T = smooth quadratic Bézier curveto
* A = elliptical Arc
* Z = closepath

**Note:** All of the commands above can also be expressed with lower letters. Capital letters means absolutely positioned, lower cases means relatively positioned.

Example 1

The example below defines a path that starts at position 150,0 with a line to position 75,200 then from there, a line to 225,200 and finally closing the path back to 150,0:

Мы можем менять наше изображение как угодно. К примеру мы удалим данные 2 линии и добавим текст.  
Изменим цвет нашего пакмена на ему соответвующий.

At some point, you’ll want to embed your finely-crafted SVG directly into a web page. There are no less than six ways to achieve that goal — but not all methods are created equally.

## 1. Using an <object> Tag

If you intend using any advanced SVG features such as CSS and scripting, the HTML5 <object> tag is your best option:

## 2. Using an <embed> Tag

I’m including <embed> for the purpose of completeness but don’t use it! While it’s similar to <object>, <embed> never has been and probably never will be part of any HTML or XHTML specification. However, it’s supported by most browsers and is often used to implement Flash plugins.

Here’s the code. It works, but don’t use it!…

## 3. Within an <iframe>

Since browsers can render SVG documents in their own right, it’s possible to load images within an iframe:

This may be a good method if you want to completely separate SVG code and script from your main page. However, manipulating an SVG image from your main page’s JavaScript will become a little more difficult.

Personally, I prefer to avoid iframes but that’s not to say you should never use them.

## 4. Inline SVG XML Embedded Into Your HTML5 Page

An SVG image can be added as a code island directly within your

The method works in all HTML5 browsers and also permits animation, scripting and CSS. But should you use it? The method feels a little dirty to me. It’s possibly an option for simple images or if you’re building the image using JavaScript, but I recommend you keep your files separate when possible.

## 5. Using an <img> Tag

The usual width, height, alt and other attributes can be added should you require them.

You’re probably wondering why img isn’t #1 in this list. For security reasons, browsers will disable SVG scripts, linking and other types of interactivity when they’re added to your page with an img tag. In addition, IE9, Chrome and Safari won’t apply stylesheet rules to the SVG if they’re defined in a separate CSS file.

## 6. Using a CSS Background Image

Like <img> tags, SVG scripts, linking and other types of interactivity are disabled when you use this method.

## Which Should You Use?

I recommend using the <object> tag to display SVG images. It feels a little unnatural, but it’s the most reliable method if you want to provide dynamic effects.

For images without interaction, the <img> tag or a CSS background can be used.

Inline SVGs or iframes are possibly options for some projects, but it’s best to avoid <embed>.

В дальнийших примерах мы будем использовать <object> tag

Inner style

Для задания внутренних стилей необходимо модифицировать сам svg документ, как показано здесь.

Внешние стили

для использования внешних ,в случае использования <object> стилей необходимо указать путь к файу со стилями в файле svg.

JS  
при манипульровании изображением через js мы подключаем js в нашем html и далее работаем с DOM, через id получаем object и работаем с DOM Svg. Где устанавливаем аттрибуты нашим path.  
В данном примере мы установили заливку желтую пакмену и прозрачную заливку тексту,а так же задали stroke обозначив width and color;