

SVG

SVG (Scalable Vector Graphics = Gráficos Vectoriales Escalables) es un lenguaje de marcas creado por el W3C y dirigido a la representación de gráficos vectoriales (dibujos y texto).

En un gráfico vectorial, los elementos de la imagen están definidos como formas elementales (líneas, rectángulos, círculos, curvas, polígonos, etc.). Basado en XML (eXtensible Markup Language). SVG no es un formato adecuada para fotografías, pero es idóneo para cualquier tipo de dibujo, técnico o artístico.

Las ventajas de SVG son muchas:

- Las imágenes SVG se pueden ampliar a cualquier escala sin perder calidad, ya que están definidas como formas que el navegador dibuja con la precisión necesaria.
- Las imágenes SVG suelen ocupar poco espacio, ya que están definidas mediante etiquetas. El tamaño en KB de la imagen es además independiente del tamaño con el que se ve en la página web.
- Las imágenes se pueden reutilizar y combinar fácilmente ya que basta con copiar el código fuente de una imagen a otra.
- Las imágenes se pueden modificar de forma dinámica mediante hojas de estilo o javascript porque forman parte de la página web.

Los elementos SVG pertenecen al **DOM** (*Document Object Model*). Por ejemplo: si queremos dibujar un círculo, el círculo es en realidad una etiqueta (*tag*) `<circle>`, y un `<script>` puede referirse al círculo y lo puede manipular, casi de la misma manera como puede manipular un `<div>`.

Podemos crear una imagen SVG de dos formas: mediante código o con la ayuda de software de dibujo vectorial (Inkscape, illustrator, ...)

Etiqueta `<svg>`

La etiqueta `svg` nos permite crear imágenes bidimensionales en formato `svg`.

Es el contenedor del código de los distintos elementos que forman la imagen, además define las dimensiones del lienzo (`viewport`), y también podemos definir la caja de visualización (`viewBox`).

En la etiqueta SVG podemos escribir atributos como las dimensiones `viewport` con `width` y `height` y la `viewBox` y algunos más. Después, van las etiquetas que describen la imagen, como `line`, `circle`, `rect`, Y terminamos con `</svg>`

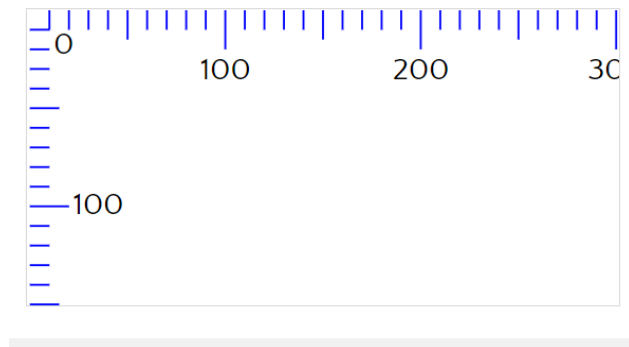
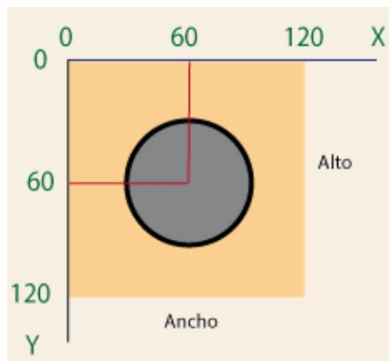
Ejemplo;

```
<svg width="120" height="120">
```

```
<circle cx="60" cy="60" r="50" stroke="black" stroke-width="4" fill="grey" />
```

```
</svg>
```

Sistema de coordenadas



El origen de coordenadas es el punto (0, 0) situado en la esquina superior izquierda del lienzo.

Los valores de X crecen hacia la derecha. Los valores de Y crecen hacia abajo.

Cuando situamos un objeto lo hacemos respecto al origen (0, 0). Por tanto, en nuestro ejemplo estamos indicando que el centro del círculo se sitúe en el punto 60, 60. Además indicamos que tiene un radio de 30.

A tener en cuenta: si no especificamos unidad de medida por defecto son píxeles. Podemos utilizar cualquier unidad de medida absoluta o relativa que conozcamos (% , em, ...)

Los gráficos SVG son elementos inline, es decir, se colocan uno al lado del otro sin salto de línea.

Dimensiones de viewport

Se definen mediante width y height. Es el área donde podemos dibujar, establece el tamaño del lienzo. Podemos dar las dimensiones en valores absolutos o en valores relativos.

Área de visualización. ViewBox

Define el área visible dentro de viewPort mediante cuatro parámetros: min-x min-y son las coordenadas de la esquina superior izquierda, el ancho y el alto del viewBox.

Especificar el viewBox es opcional. Mediante el viewBox podemos realizar efectos de zoom, visualizar solo una parte del dibujo y otros efectos de distorsión, como vamos a ver en los siguientes ejemplos.

Atributo preserveAspectRatio

Indica cómo un elemento con un viewBox que proporciona una relación de aspecto determinada debe encajar en una viewPort con una relación de aspecto diferente.

Solo tiene sentido si se especifica el viewBox. Consta de dos valores separados por un espacio. El segundo valor es opcional. El primer valor consta de dos valores, el primero hace referencia a la coordenada x. y el segundo a la coordenada y los dos indican cómo se desplaza o alinea el viewBox respecto al lienzo del documento SVG

- xMin alinea la coordenada x del viewBox con el valor mínimo de la x en el lienzo.
- xMid alinea la coordenada x del viewBox con el valor medio de la x en el lienzo.
- xMax alinea la coordenada x del viewBox con el valor máximo de la x en el lienzo.
- YMin alinea la coordenada y del viewBox con el valor mínimo de la y en el lienzo.
- YMid alinea la coordenada y del viewBox con el valor medio de la y en el lienzo.
- YMax alinea la coordenada y del viewBox con el valor máximo de la y en el lienzo.

Ejemplo:

xMin YMin hacen que el viewBox se alinee respecto a la esquina superior izquierda del lienzo,

xMax YMax hacen que el viewBox se alinee respecto a la esquina inferior derecha del lienzo.

El segundo valor de *preserveAspectRatio* tiene tres opciones, **meet**, **slice** y **none**.

meet ajusta el viewport dentro del lienzo.

slice permite que el viewport sobrepase las dimensiones del documento.

none, distorsiona el viewport para que encaje en el documento

Ejemplos

<h6>Sin viewBox</h6>

```
<svg x="0" y="0" width="310" height="110">
  <rect width="310" height="110" fill="grey" />
  <rect x="10" y="10" width="90" height="90" fill="yellow" />
  <rect x="110" y="10" width="90" height="90" fill="green" />
  <rect x="210" y="10" width="90" height="90" fill="blue" />
```

</svg>

<h6>preserveAspectRatio none </h6>

```
<svg x="0" y="0" width="310" height="110" viewBox="0 0 310 220" preserveA
spectRatio="none">
  <rect width="310" height="110" fill="grey" />
  <rect x="10" y="10" width="90" height="90" fill="yellow" />
  <rect x="110" y="10" width="90" height="90" fill="green" />
  <rect x="210" y="10" width="90" height="90" fill="blue" />
```

</svg>

<hr>

<h6>preserveAspectRatio xMinYmax meet </h6>

```
<svg x="0" y="0" width="310" height="110" viewBox="0 0 310 220" preserveA
spectRatio="xMinYMax meet">
  <rect width="310" height="220" fill="grey" />
  <rect x="10" y="10" width="90" height="90" fill="yellow" />
  <rect x="110" y="10" width="90" height="90" fill="green" />
  <rect x="210" y="10" width="90" height="90" fill="blue" />
```

</svg>

<hr>

<h6>preserveAspectRatio xMinYmid slice </h6>

```
<svg x="0" y="0" width="310" height="110" viewBox="0 0 310 220" preserveA
spectRatio="xMinYMid slice">
  <rect width="310" height="220" fill="grey" />
  <rect x="10" y="10" width="90" height="90" fill="yellow" />
  <rect x="110" y="10" width="90" height="90" fill="green" />
  <rect x="210" y="10" width="90" height="90" fill="blue" />
```

</svg>

Dibujar formas básicas

Círculo:

```
<circle cx="0" cy="0" r="50" fill="none" stroke="red" stroke-width="1" />
```

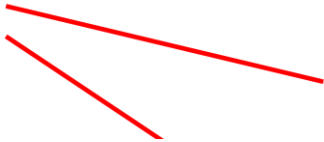
cx, cy: coordenadas centro del círculo

r: radio



Línea: dibuja una línea entre dos puntos: x1, y1; x2, y2.

```
<line x1="-200" y1="0" x2="200" y2="0" stroke="black" stroke-width="1" />
```



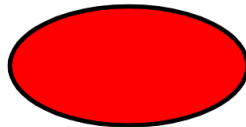
Rectángulo: dibuja un rectángulo, especificamos la esquina superior izquierda y su tamaño. En el ejemplo con las esquinas redondeadas

```
<rect x="110" y="10" width="90" height="90" fill="green" rx="20" />
```



Elipse. Especificamos coordenadas del punto central y el radio horizontal y vertical

```
<ellipse cx="110" cy="60" rx="80" ry="40" stroke="black" stroke-width="3" fill="red" />
```

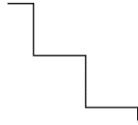


Polilínea. Dibuja una secuencia de líneas rectas que empiezan en un punto.

Especificamos una secuencia de puntos. Si el último punto no coincide con el primero tenemos una polilínea abierta, si coinciden será cerrada o, lo que es igual, un polígono.

Parámetros: x y coordenadas de cada punto de la polilínea

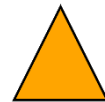
```
<polyline fill="none" stroke="black" points="20,40 40,40 40,80 80,80 80,120 120,120 120,130" />
```



Polígono. Dibuja una secuencia de líneas rectas enlazadas, definidas por un conjunto de puntos, de forma automática el último punto se une con el primero para crear un polígono.

Parámetros: **x y** coordenadas de cada punto del polígono.

```
<polygon fill="orange" stroke="black" stroke-  
width="5px" points="80 230, 180 30, 280 230" />
```



Texto en SVG

Podemos crear texto con la etiqueta text. Podemos posicionar, personalizar y aplicar propiedades CSS al texto.

```
<text x="20" y="50">Este es un texto en SVG</text>  
  <text x="20" y="100" fill="yellow" stroke="blue" stroke-  
width="1px">Este es un texto SVG</text>  
  <text x="20" y="150" fill="yellow" stroke="blue" stroke-  
width="2px" style="font-family:impact; font-size:40px; font-  
weight: 800;">Este es un texto SVG</text>
```

Este es un texto en SVG

Este es un texto SVG

Este es un texto SVG

https://developer.mozilla.org/es/docs/Learn/HTML/Multimedia_and_embedding/Adding_vector_graphics_to_the_Web

<https://www.mclibre.org/consultar/htmlcss/html/html-svg.html>