

## 03-08 etiquetas HTML svg Figuras Avanzadas

### Atributos de formato

Son opcionales y van dentro de las figuras . Si no se ponen por defecto salen negros

- fill ="color del borde, transparente o null"
- = " color del borde" si no pones el siguiente no se pinta nada
- stroke-width ="anchura del trazo"

por ejemplo una circunferencia naranja y otro dentro naranja oscuro

```
<svg viewBox="-100 -100 310 310" xmlns="http://www.w3.org/2000/svg"
width="200" height="200">
<circle cx="100" cy="100" r="20" stroke="#ffff77" fill="transparent" stroke-width="3"/>
<circle cx="100" cy="100" r="40" stroke="#ffff00" fill="transparent" stroke-width="6"/>
<circle cx="100" cy="100" r="60" stroke="#EEEE00" fill="transparent" stroke-width="9"/>
<circle cx="100" cy="100" r="80" stroke="orange" fill="transparent" stroke-width="12"/>
<circle cx="100" cy="100" r="100" stroke="darkorange" fill="transparent" stroke-width="15"/>
</svg>
```

### Más Figuras

Para insertar una forma, crea un elemento en el documento. Diferentes elementos corresponden a diferentes formas y toman diferentes parámetros para describir el tamaño y la posición de esas formas. Algunos son un poco redundantes porque pueden crearse con otras formas, pero todos están ahí para su conveniencia y para mantener sus documentos SVG lo más breves y legibles posible.

### Rectángulo

El elemento <rect> dibuja un rectángulo en la pantalla. Hay 6 atributos básicos que controlan la posición y la forma de los rectángulos en la pantalla. El de la derecha tiene configurados sus parámetros rx y ry, lo que le da esquinas redondeadas. Si no están configurados, por defecto son 0.

```
<rect x="10" y="10" width="30" height="30"/>
<rect x="60" y="10" rx="10" ry="10" width="30" height="30"/>
```

X	La posición x de la esquina superior izquierda del rectángulo.
y	La posición y de la esquina superior izquierda del rectángulo.
width	El width del rectángulo
height	La height del rectángulo
ry	El radio x de las esquinas del rectángulo.
rx	El radio y de las esquinas del rectángulo.

```
<svg width="120" height="120" viewBox="0 0 120 120" version="1.1" xmlns="http://www.w3.org/2000/svg" width="200" height="200">
<rect x="10" y="10" width="100" height="100"/>
</svg>
```

## Círculo

El elemento <circle> dibuja un círculo en la pantalla. Se necesitan 3 parámetros básicos para determinar la forma y el tamaño del elemento.

```
<círculo cx="25" cy="75" r="20"/>
```

r El radio del círculo.

cx La posición x del centro del círculo.

cy La posición y del centro del círculo.

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg"
      width="200" height="200" viewBox="-100 -100 200 200">
  <circle cx="0" cy="0" r="75" fill="none" stroke="red" stroke-width="5" />
  <circle cx="0" cy="0" r="50" fill="none" stroke="blue" stroke-width="5" />
  <circle cx="0" cy="0" r="25" fill="none" stroke="green" stroke-width="5" />
</svg>
```

```
<svg version="1.1" xmlns="http://www.w3.org/2000/svg"
      width="100" height="100"
      style="background-color: lightgray">
  <circle cx="50" cy="50" r="50" fill="none" stroke="red" stroke-width="1" />
</svg>
<br>
```

## Elipse

Una <ellipse> es una forma más general del elemento <circle>, donde puede escalar el radio x e y (comúnmente denominados ejes semimayor y semiminor en matemáticas) del círculo por separado.

```
<ellipse cx="75" cy="75" rx="20" ry="5"/>
```

- rx El radio x de la elipse.
- ry El radio y de la elipse.
- cx La posición x del centro de la elipse.
- cy La posición y del centro de la elipse.

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg" width="200" height="200" >
  <ellipse cx="100" cy="50" rx="100" ry="50" />
</svg>
```

```
<svg viewBox="-100 -100 310 310" xmlns="http://www.w3.org/2000/svg" width="200"
      height="200" >
  <ellipse cx="75" cy="75" rx="10" ry="50" stroke="orange" fill="transparent" stroke-width="3"/>
  <ellipse cx="75" cy="75" rx="20" ry="40" stroke="orange" fill="transparent" stroke-width="3"/>
  <ellipse cx="75" cy="75" rx="30" ry="30" stroke="orange" fill="transparent" stroke-width="3"/>
  <ellipse cx="75" cy="75" rx="40" ry="10" stroke="orange" fill="transparent" stroke-width="3"/>
```



# Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

```
<ellipse cx="75" cy="75" rx="50" ry="0" stroke="orange" fill="transparent" stroke-width="3"/>
```

```
</svg>
```

## line

El elemento `<line>` toma las posiciones de dos puntos como parámetros y dibuja una línea recta entre ellos.

```
<line x1="10" x2="50" y1="110" y2="150" stroke="black" stroke-width="5"/>
```

`x1` --> La posición x del punto 1.

`y1` --> La posición y del punto 1.

`x2` --> La posición x del punto 2.

`y2` --> La posición y del punto 2.

## polyline

Una `<polyline>` es un grupo de líneas rectas conectadas. Dado que la lista de puntos puede ser bastante larga, todos los puntos se incluyen en un atributo:

```
<polyline points="60, 110 65, 120 70, 115 75, 130 80, 125 85, 140 90, 135 95, 150 100, 145"/>
```

`points` --> Una lista de puntos. Cada número debe estar separado por un espacio, coma, EOL o un carácter de sheight de línea.

Cada punto debe contener dos números: una coordenada x y una coordenada y. Entonces, la lista (0,0), (1,1) y (2,2) se escribiría como 0, 0 1, 1 2, 2.

### Ejemplo

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg" >
< polyline points="0,100 50,25 50,75 100,0" fill="red" />
< polyline points="100,100 150,25 150,75 200,0" fill="none" stroke="black" />
</svg>
```

## polygon

Un `<polygon>` es similar a una `<polyline>`, ya que se compone de segmentos de línea recta que conectan una lista de puntos. Sin embargo, para los polígonos, la ruta conecta automáticamente el último punto con el primero, creando una forma cerrada.

Nota: un rectángulo es un tipo de polígono, por lo que se puede usar un polígono para crear un elemento `<rect>` en los casos en que necesite un poco más de flexibilidad.



# Diseño de Interfaces Web

Ciclo de Grado Superior Desarrollo de Aplicaciones Web

IES Playamar

```
<polygon points="50, 160 55, 180 70, 180 60, 190 65, 205 50, 195 35, 205 40, 190 30, 180 45, 180"/>
```

points --> Una lista de puntos, cada número separado por un espacio, coma, EOL o un carácter de avance de línea. Cada punto debe contener dos números: una coordenada x y una coordenada y. Entonces, la lista (0,0), (1,1) y (2,2) se escribiría como 0, 0 1, 1 2, 2. Luego, el dibujo cierra el camino.

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg">
<polygon points="0,100 50,25 50,75 100,0" />
<!-- Example of the same polygon shape with stroke and no fill --&gt;
&lt;polygon points="100,100 150,25 150,75 200,0" fill="none" stroke="black" /&gt;
&lt;/svg&gt;</pre>
```

## path

Una <ruta> es la forma más general que se puede usar en SVG. Usando un elemento de ruta, puede dibujar rectángulos (con o sin esquinas redondeadas), círculos, elipses, polilíneas y polígonos. Básicamente cualquiera de los otros tipos de formas, curvas bezier, curvas cuadráticas y muchas más.

Ejemplo:

```
<path d="M20,230 Q40,205 50,230 T90,230" fill="none" stroke="blue" stroke-width="5"/>
```

d --> Una lista de puntos y otra información sobre cómo dibujar la ruta. Consulte la sección Rutas para obtener más información.

Ejemplo

```
<svg viewBox="0 0 200 100" xmlns="http://www.w3.org/2000/svg" width="200" height="200" >
<path fill="none" stroke="lightgrey" d="M20,50 C20,-50 180,150 180,50 C180-50 20,150 20,50 z" />
</svg>
```

Usa la herramienta <https://yqnn.github.io/svg-path-editor/> para conseguir los datos del path sencillamente

M 417 -83 C 475 -90 484 -6 445 62 C 385 145 250 -180 195 118 C 177 220 81.6667 -7.3333 274 60  
C 335 79 331 -78 415 -82

## Practica 03-08

**Ejercicio 1.** Crea cada uno de los ejemplos que aparecen el la practica

**Ejercicio 2.** Crea un svg que aparezca este diseño con círculos

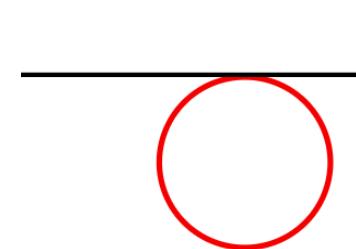
(ojo el centro es el mismo el stroke tiene un degradado de amarillos desde el DD8800 hasta el FFFF00 en el mas pequeño y los stroke-width bajan de 3 en tres del mas externo al interno)



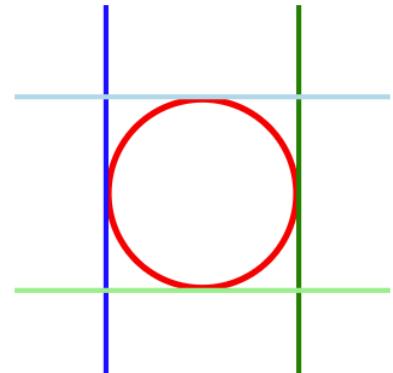
**Ejercicio 3.** Crea un svg con dos lineas perpendiculares y paralelas



Ejercicio 3 Paralela y perpendicular



Ejercicio 4



Ejercicio 5

**Ejercicio 6.** Crea un svg que aparezca este diseño con elipses. Tienes que hacer una progresión desde los valores

`cx="75" cy="75" rx="10" ry="50"`

Hasta los valores

`cx="75" cy="75" rx="50" ry="0"`

sumando o restando de 10 en 10

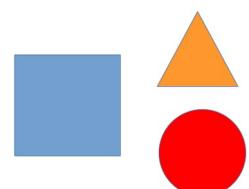
El estilo que tiene es el siguiente:

`stroke="orange" fill="transparent" stroke-width="3"`

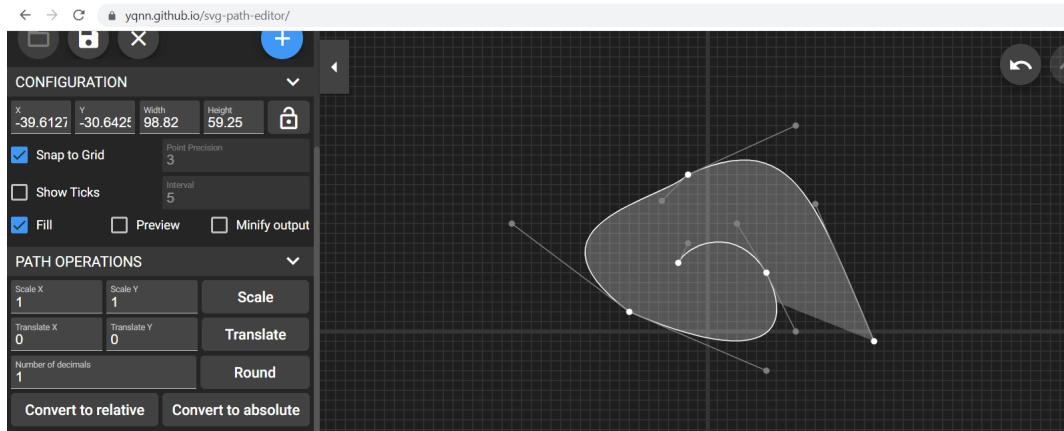


**Ejercicio 7.** Crea un svg con esta forma (mas o menos)

el triangulo es un polígono de tres puntos



Ejercicio 8. Usa <https://yqnn.github.io/svg-path-editor/> para generar una ruta chula



ejemplo de path

```
M -3 -7 C -2 -9 3 -11 6 -6 C 9 0 6 4 -8 -2 C -20 -11 -4.6667  
-13.3333 -2 -16 C 9 -21 11 -13 17 1
```

```
M 117 -83 C 175 -90 184 -6 145 62 C 85 145 -50 -180 -105 118 C -123 220 -218.3333 -7.3333 -26 60  
C 35 79 31 -78 115 -82z
```