

**SVG**

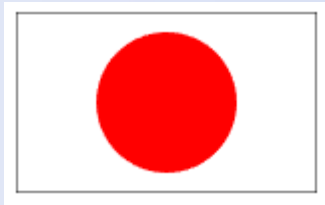
# ¿Qué es?

- Scalable Vector Graphics, SVG, es un lenguaje de marcado XML W3C para la generación de gráficos. Es parcialmente implementado en los principales navegadores.
- Con el elemento SVG (Scalable Vector Graphics) podemos crear imágenes vectoriales redimensionables de manera dinámica o interactiva utilizando diferentes etiquetas. Programas como Inkscape permiten guardar en dicho formato.
- Entre otras etiquetas, están:
  - rect (para crear rectángulos)
  - circle (círculos)
  - line
  - ellipse
  - polyline (multi-líneas)
  - polygon (camino cerrado de líneas)
  - path (camino cerrado con todo tipo de líneas y curvas)
  - text

# Ejemplos simples

- ```
<svg width="150" height="90" xmlns="http://www.w3.org/2000/svg">  
  <rect width="150" height="90" fill="white" stroke="black" stroke-width="1"/>  
  <circle cx="75" cy="45" r="35" fill="red"/>  
</svg>
```
- ```
<svg width="150" height="90" xmlns="http://www.w3.org/2000/svg">  
  <rect width="150" height="90" fill="green" stroke="black" stroke-width="1"/>  
  <rect x="50" y="1" width="50" height="90" fill="white"/>  
  <rect x="100" y="1" width="50" height="90" fill="red"/>  
</svg>
```
- ```
<rect width="150" height="90" fill="red" stroke="black" stroke-width="1"/>  
<rect x="38" y="0" width="30" height="90" fill="white"/>  
<rect x="0" y="33" width="150" height="30" fill="white"/>  
<rect x="45" y="0" width="16" height="90" fill="blue"/>  
<rect x="0" y="40" width="150" height="16" fill="blue"/>
```

# Ejemplos simples



# Mover objetos

- Si tras acabar un objeto descubriéramos que hay que moverlo, en vez de mover todas sus coordenadas podemos aplicar un transform:
- `<text x="30" y="35" rotate="17 0 -9 11 10 -3" fill="red" stroke="white" stroke-width="2" transform="translate(20,5)">`

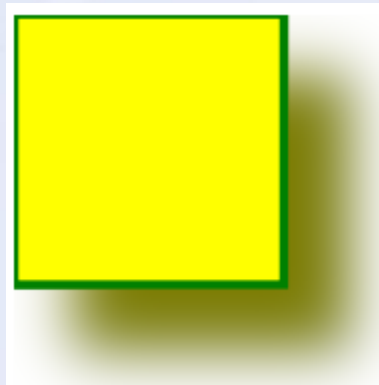
Texto de prueba

`</text>`

# Filtros

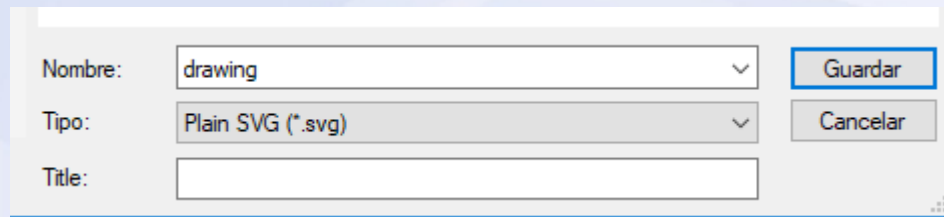
- Utilizando la etiqueta `<defs>` seguida de `<filter>` podemos utilizar distintos tipos de filtros.

Ver ejemplos adjuntos.



# Una aplicación: iconos

- Puedes usar un programa de dibujo vectorial para hacer un icono o un logotipo.
- Al guardarlo, y seleccionar svg, en lugar de generar un archivo, indicar que te dé el código o que lo guarde en formato SVG plano, para obtener una versión reducida de texto plano del mismo. Por ejemplo, en Inkscape:



- Una vez tengas el código, ya puedes integrarlo en la página y hacer que interactúe con CSS.
  - Cambios en color, tamaño, perspectiva... eventos.

# Una aplicación: iconos

- ¿Por qué no Icon Fonts? Aunque los “icon fonts” están bien para una implementación rápida de iconos sencillos, tienen algunas deficiencias que no tienen los SVG:
  - Sus etiquetas utilizadas no son semánticas (por ejemplo `<span>`).
  - Son monocromo y no pueden ser multicoloreados.
  - Tienen fallos extraños en muchos navegadores, relacionados con filtros anti-aliasing porque los consideren como texto. También puede ser difícil posicionarlos.
- Las ventajas de iconos SVG son, entre otras:
  - SVGs son más semánticos. Un SVG es una imagen y el icono es marcado así como una.
  - Los iconos SVG se ven nítidos y claros en cualquier resolución de pantalla.
  - Te proveen de más control sobre los estilos, permitiendo entre otros iconos multicolor. Partes individuales del SVG pueden ser seleccionadas y estilizadas usando CSS.
  - Los iconos SVG pueden ser animados e interactivos. Pueden incluir “morphing” y otro tipo de animaciones complejas que no son posibles en los “icon fonts”.
  - SVGs son accesibles.



# Otra aplicación: gráficas

- ¿Por qué no CSS?

- CSS puede ser utilizado para crear gráficas simples que requieren formas “regulares”, como rectángulos y círculos. Las gráficas de barras horizontales o verticales son un excelente ejemplo de gráficas sólo CSS. El mejor modo además es poner los datos en un `<table>` con lo que el dato queda marcado de un modo semántico y puede ser modificado desde el lado del servidor, como PHP.
- Nunca va a ser tan interactivo, visualizable y accesible como SVG, en particular para formas irregulares.

# Gráficas

- ¿Por qué no Canvas?

- HTML5 Canvas también puede ser usado para realizar gráficas, pero el contenido del Canvas no es parte del DOM y no es accesible por lectores de pantalla. Necesitarías crear un contenido secundario entre las etiquetas de apertura y cierre de `<canvas>` para servir como información del contenido y hacerlo accesible. También mapear el contenido alternativo con el del Canvas para mostrar lo que habrá en su lugar → doble de trabajo.
- Con SVG obtienes accesibilidad y contenido semántico así como interactividad con JavaScript sencilla.
- Aun así para Canvas hay alternativas como usar el plugin Visualize en JQuery que toma los datos de un elemento `<table>`.

# Gráficas

- ¿Por qué SVG?

El formato de imagen SVG tiene varias ventajas, entre las que están:

- Ficheros de tamaño pequeño que se comprimen bien.
- Escala a cualquier tamaño sin perder claridad.
- Se ve perfectamente en pantallas de tipo retina.
- Se le puede aplicar interactividad y filtros.

- Además:

- SVGs son accesibles para lectores de pantalla (con un poco de trabajo).
- Hay muchos Frameworks basados en SVG.

# Gráficas

- Como primera aproximación, podríamos usar una etiqueta del tipo `` para introducir una gráfica previamente creada en un programa de dibujo.
- Perderíamos interactividad y accesibilidad.
- Por tanto lo mejor es usar una combinación de JavaScript, CSS y HTML con el elemento `<SVG>`
  - Tener en cuenta la etiqueta `<g>` para agrupar elementos relacionados.

# Gráficas

- Podemos ver distintos tipos de gráficas en ejemplos adjuntos.

# Ejercicios

- **Ejercicio 1:** Genera o busca una imagen SVG y aplícale un filtro como el de la sombra.
- **Ejercicio 2:** Busca un icono ya realizado en internet y personalízalo. Ten en cuenta webs como <https://www.flaticon.com/>
- **Ejercicio 3:** Realiza un gráfico circular para un conjunto de datos dado desde el servidor, reutilizando código ya visto y cambiándole los estilos.
- **Ejercicio 4:** Realiza una gráfica de barras utilizando un framework que utilice SVG, JavaScript y CSS.

# Ejercicios

- **Debate en clase:** Ventajas y desventajas del uso de frameworks. Cuándo usarlos y cuándo no.

# Ejercicios

- **Ejercicio 5:** Aplícale algún tipo de animación al icono, cambio de color, etc., cuando algo ocurra en la pantalla (ejemplo, :hover).