

# **Efectos gráficos con SVG**

# Transparencias

- Con HTML-SVG tenemos el atributo opacity, que podemos usar (igual que el opacity de CSS3):

```
<svg width=270 height=270>
```

```
<circle cx=70 cy=70 r=60 stroke="yellow" stroke-width=5 fill="blue"
opacity=0.5 />
```

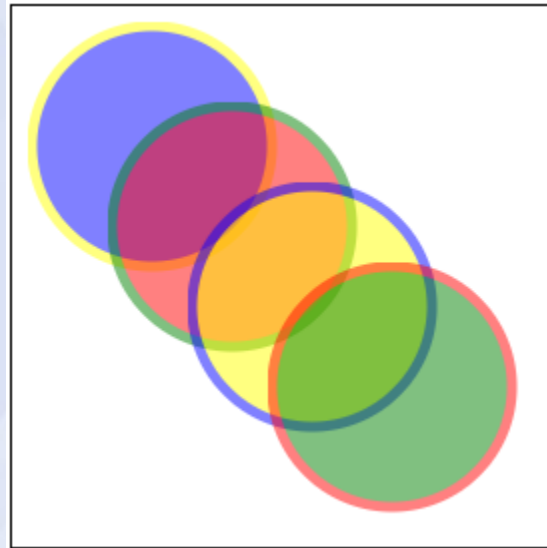
```
<circle cx=110 cy=110 r=60 stroke="green" stroke-width=5 fill="red"
opacity=0.5 />
```

```
<circle cx=150 cy=150 r=60 stroke="blue" stroke-width=5 fill="yellow"
opacity=0.5 />
```

```
<circle cx=190 cy=190 r=60 stroke="red" stroke-width=5 fill="green"
opacity=0.5 />
```

```
</svg>
```

# Transparencias



- También podríamos haber usado los atributos `fill-opacity` y `stroke-opacity` para afectar sólo al fondo o al borde, respectivamente. También existen en CSS3.

# Degradados

- Un gradiente de color, también llamado degradado de color, es una transición de manera suave desde un color a otro, pasando por todas las gamas intermedias entre ellos.
- En SVG tenemos dos tipos de gradientes, los gradientes lineales y los gradientes radiales.
- Los gradientes pueden aplicarse a cualquier elemento al que le hayamos dado color, tanto en su relleno (fill) como en el trazado del borde (stroke).

# Degradado lineal

- Para crear un gradiente lineal utilizaremos la etiqueta `linearGradient`, dentro de la etiqueta `defs`:

```
<defs>
  <linearGradient ..... >
    .....
  </linearGradient>
</defs>
```

- La etiqueta `<defs> ...</defs>` no incluye ningún tipo de elemento en la página, y sólo sirve para organizar el contenido. Incluimos en esta etiqueta una serie de etiquetas que sirven para definir algunos elementos que incluiremos más adelante en la página, y que enlazaremos mediante el id.

# Degradado lineal

- Ejemplo:

```
<svg width=250 height=250>  
<defs>  
  <linearGradient id="grad1"  
    x1="50%" y1="0%" x2="50%" y2="100%">  
    <stop offset="0" stop-color="#2222FF"/>  
    <stop offset="0.2" stop-color="#1111AA"/>  
    <stop offset="0.5" stop-color="#111188"/>  
    <stop offset="0.9" stop-color="111144"/>  
    <stop offset="1" stop-color="silver"/>  
  </linearGradient>  
</defs>  
<circle cx=125 cy=125 r=120 fill="url(#grad1)"/>  
</svg>
```

# Degradado lineal



# Degradado lineal

- Atributos: (1/2)

- `id="<nombre_id>"` : Como valor llevará un nombre identificador para poder referirse a este degradado en otro elemento. Es por tanto obligatorio.
- `x1="<medida>" y1="<medida>"` : Indican las coordenadas "x" e "y" del punto en donde se inicia el degradado de color. Si no indicamos la medida (y escribimos sólo el número), se entiende que las medidas son en píxeles. También podemos poner porcentajes. En el ejemplo de la circunferencia hemos puesto X al 50% en inicio y fin para que empiece y acabe en los "polos", o de arriba a abajo.
- `x2="<medida>" y2="<medida>"` : Indican las coordenadas "x" e "y" del punto en donde termina el degradado de color. Al igual que en los anteriores podemos poner sólo el número (pixels) o porcentajes.
- `gradientUnits="userSpaceOnUse | objectBoundingBox"` : desde dónde se miden las coordenadas. Con el valor "userSpaceOnUse" las coordenadas son las mismas que para el elemento contenedor del SVG, mientras que `objectBoundingBox` indica que toma como origen de coordenadas el propio elemento al que se le aplica, en su esquina superior derecha de la caja (imaginaria) que lo contiene. Este último es el valor por defecto, y el que se aplica si el atributo no se pone.



# Degradado lineal

- Atributos (2/2):

- `spreadMethod = "pad | reflect | repeat"` : Indica cómo debe tratarse el resultado del degradado. El valor `pad` es el valor por defecto y hace que antes y después de las coordenadas indicadas, los colores se extiendan de manera uniforme (sin degradado). El valor `reflect` repite el degradado pero en cada repetición hace un reflejo del mismo como en un espejo. El valor `repeat` repite el degradado tal cual, sin reflejarlo en cada repetición.
- `stop`: esta etiqueta tiene dos atributos, `offset` y `stop-color`. En `offset` indicamos con un número del 0 al 1 la parada en el degradado, y en `stop-color`, el nuevo color que empieza.

- Fuera ya de lo que es el degradado, pondremos el siguiente atributo a un elemento para aplicarle el degradado:  
`fill="url(#nombre_id)"`

# Degradado radial

- En el degradado o gradiente radial el efecto de cambio de color se produce mediante círculos concéntricos. El cambio de color va desde un punto central hacia el exterior formando círculos. Utilizamos la etiqueta:

```
<defs>  
  <radialGradient ..... >  
    .....  
  </linearGradient>  
</defs>
```

- La etiqueta `<defs> ...</defs>` no incluye ningún tipo de elemento en la página, y sólo sirve para organizar el contenido. Incluimos en esta etiqueta una serie de etiquetas que sirven para definir algunos elementos que incluiremos más adelante en la página, y que enlazaremos mediante el id.

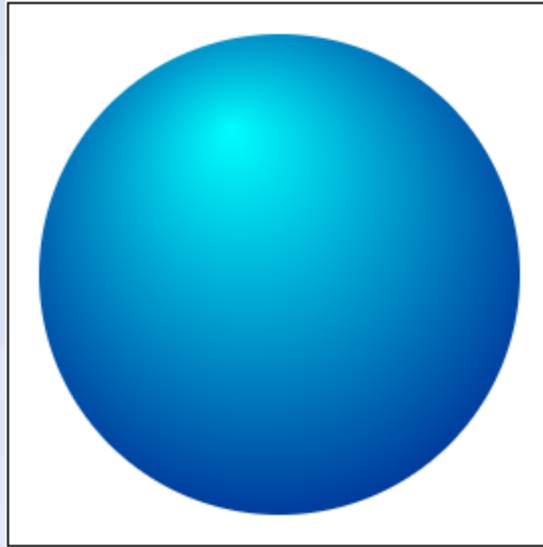
# Degradado radial

- Ejemplo:

```
<svg width=270 height=270 ><defs>  
  <radialGradient id="rad1"  
    cx="50%" cy="50%" r="75%" fx="40%" fy="20%">  
    <stop offset="0" stop-color="aqua"/>  
    <stop offset="1" stop-color="navy"/>  
  </radialGradient>  
</defs>  
<circle cx=135 cy=135 r=120 fill="url(#rad1)"/>  
</svg>
```

# Degradado radial

- Ejemplo:



# Degradado radial

- Atributos:
- `cx="<medida>" cy="<medida>"` : Coordendas "x" e "y" del centro del círculo de degradado. En estos atributos y los siguientes podemos indicar una medida, un número (píxeles) o un porcentaje.
- `r="<medida>"` : Indica el radio dentro del cual tendrá alcance el degradado.
- `fx="<medida>" fy="<medida>"` : indica el foco del degradado. El foco es el punto desde el cual se inicia el degradado. Éste se extiende desde el foco hacia el resto del círculo marcado. El foco no tiene porqué ser el mismo punto que el centro del círculo. En caso de serlo podemos omitir estos atributos.
- El resto de atributos son los mismos que para `linearGradient` y con los mismos valores, es decir `gradientUnits` y `spreadMethod`.
- Al igual que con `linearGradient` definimos aquí también los colores y transparencias con las etiquetas `<stop offset ...>` que se emplean exactamente igual. En el atributo `offset` el número 0 será el punto del foco y el número 1 será el exterior del radio.

# Incluir imágenes

- Para incluir imágenes en un contenedor SVG utilizaremos la etiqueta `image` la cual tiene varios atributos:
- `xlink:href="<url>"` : Incluimos aquí la ruta de la imagen que queremos insertar.
- `x="<medida>"`      `y="<medida>"` : Las coordenadas "x" e "y" del sitio en el que se inserta la imagen (punto superior izquierdo de la imagen).
- `width="<medida>"`    `height="<medida>"` : El ancho y el alto de la imagen que se va a insertar.

# Incluir imágenes

- Ejemplo:

```
<svg width=270 height=220>
```

```
<rect      x=0      y=120      width=270      height=100  
fill="#ffb6b7"/>
```

```
<image  xlink:href="objetos/estrella.gif"  x=40  y=10  
width=150 height=50 />
```

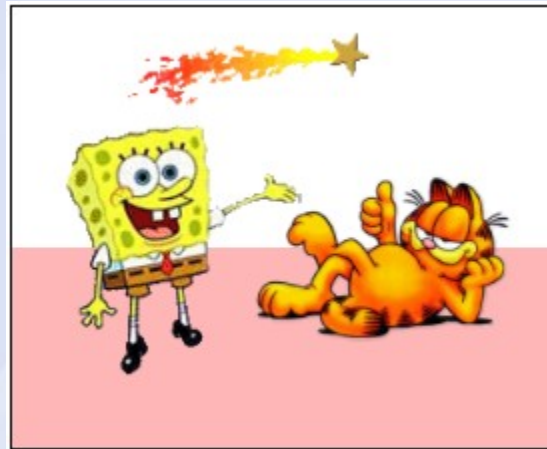
```
<image      xlink:href="objetos/bobesponja.gif"      x=10  
y=50 width=150 height=135 />
```

```
<image  xlink:href="objetos/garfield.gif"  x=120  y=80  
width=130 height=90 />
```

```
<svg>
```



# Incluir imágenes





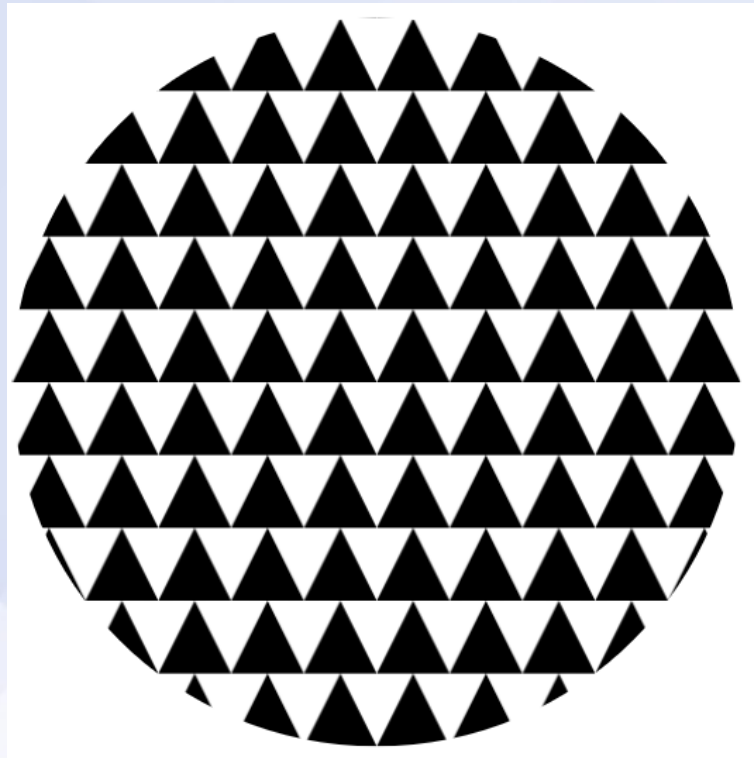
# Texturas o patrones

- Podemos añadir texturas fácilmente mediante el elemento `<pattern>`.
- Este elemento define un objeto gráfico que puede ser redibujado y repetido en intervalos de coordenadas x e y para cubrir un área.
- El `<pattern>` es referenciado por los atributos `fill` y `stroke`.
- Podemos hacer patrones con elementos SVG o con imágenes.

# Ejemplo de pattern con SVG

```
<svg width="620" height="620" viewBox="0 0 120 120"
xmlns="http://www.w3.org/2000/svg">
  <defs>
    <pattern id="Triangle" width="10" height="10"
      patternUnits="userSpaceOnUse">
      <polygon points="5,0 10,10 0,10"/>
    </pattern>
  </defs>
  <circle cx="60" cy="60" r="50" fill="url(#Triangle)"/>
</svg>
```

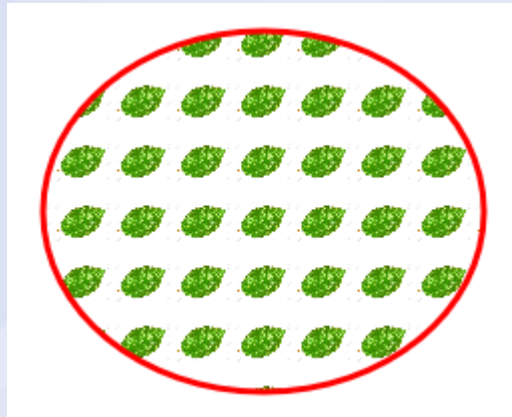
# Ejemplo de pattern con SVG



# Ejemplo de pattern con imagen

```
<svg width="270" height="200" >
<defs>
<pattern id="patron1" x="0" y="0" width="30" height="30"
patternUnits="userSpaceOnUse" viewBox="0 0 50 50" >
<image
xlink:href="https://previews.123rf.com/images/scis65/scis651305/scis651
30500036/19873707-Fondo-verde-de-la-hoja-rosa-Foto-de-archivo.jpg"
x="0" y="0" width="50" height="50" >
</pattern>
</defs>
<ellipse cx="135" cy="100" rx="110" ry="90"
stroke="red" stroke-width="3" fill="url(#patron1)"/>
</svg>
```

# Ejemplo de pattern con imagen



# Transformaciones

- SVG soporta el atributo transform, de similar modo a la propiedad de CSS3.

# Transformaciones

- Traslación:

```
<svg width="270" height="200">
```

```
<g>
```

```
  <rect x="10" y="10" width="80" height="80"
  fill="orange"/>
```

```
  <text x="20" y="50" style="font: bold 0.8em arial; fill:
  purple;">
```

```
    Original</text>
```

```
</g>
```

```
<g transform="translate(150,80)">
```

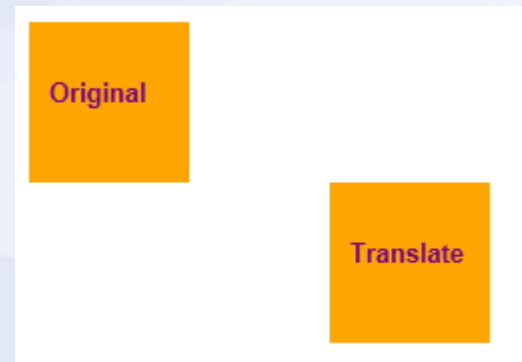
```
  <rect x="10" y="10" width="80" height="80"
  fill="orange"/>
```

```
  <text x="20" y="50" style="font: bold 0.8em arial; fill:
  purple;">
```

```
    Translate</text>
```

```
</g>
```

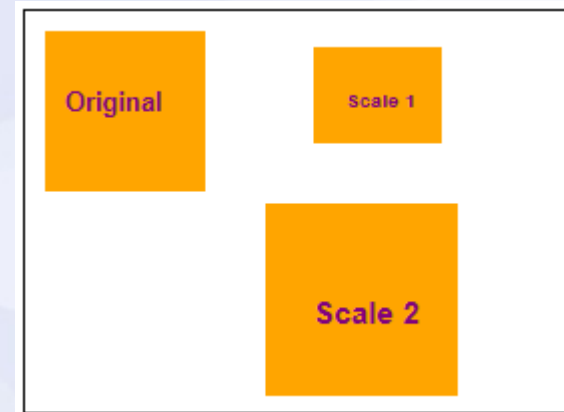
```
</svg>
```



# Transformaciones

- Escalado:

```
<svg width="270" height="200">
<g>
<rect x="10" y="10" width="80" height="80" fill="orange"/>
<text x="20" y="50" style="font: bold 0.8em arial; fill: purple;">
  Original</text>
</g>
<g transform="scale(0.8,0.6)">
  <rect x="180" y="30" width="80" height="80" fill="orange" />
  <text x="200" y="80" style="font: bold 0.8em arial; fill: purple;">
    Scale 1</text>
</g>
<g transform="scale(1.2)">
  <rect x="100" y="80" width="80" height="80" fill="orange" />
  <text x="120" y="130" style="font: bold 0.8em arial; fill: purple;">
    Scale 2</text>
</g>
</svg>
```





# Transformaciones

- Rotación:

```
<svg width="270" height="150">
```

```
<g >
```

```
    <rect x="20" y="25" width="80" height="80"
    fill="orange"/>
```

```
    <text x="30" y="75" style="font: bold 0.8em arial; fill:
    purple;">
```

```
    Original</text>
```

```
</g>
```

```
<g transform="translate(190,75),rotate(60)">
```

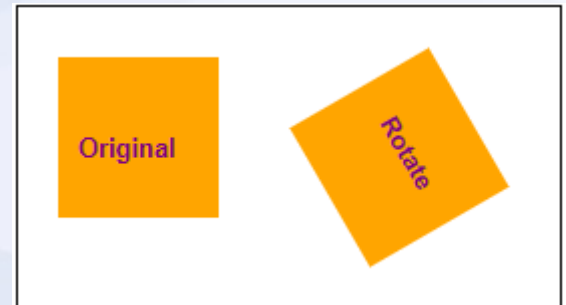
```
    <rect x="-40" y="-40" width="80" height="80"
    fill="orange"/>
```

```
    <text x="-20" y="0" style="font: bold 0.8em arial; fill:
    purple;">
```

```
    Rotate</text>
```

```
</g>
```

```
</svg>
```



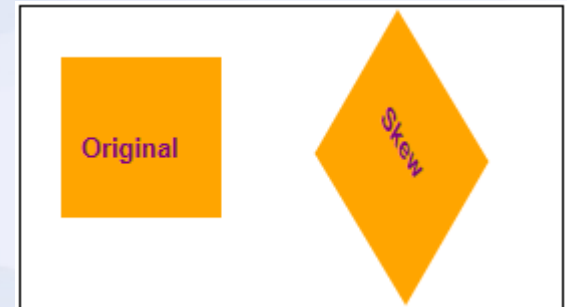
- Nota: podemos aplicar dos transformaciones al mismo tiempo separándolas por comas:

```
transform="translate(
190,75),rotate(60)"
```

# Transformaciones

- Sesgado:

```
<svg width="270" height="150">  
<g >  
  <rect x="20" y="25" width="80" height="80" fill="orange"/>  
  <text x="30" y="75" style="font: bold 0.8em arial; fill: purple;">  
    Original</text>  
</g>  
<g  
transform="translate(190,75),rotate(45),skewX(15),skewY(15)">  
  <rect x="-40" y="-40" width="80" height="80" fill="orange"/>  
  <text x="-20" y="0" style="font: bold 0.8em arial; fill: purple;">  
    Skew</text>  
</g>  
</svg>
```



# Transformaciones

- El método matrix:

El método matrix es una combinación de los métodos de traslación, de escalado y de sesgado. Tiene seis parámetros o números dentro del paréntesis.

- `transform="matrix(a,b,c,d,e,f)"`

- Los dos últimos parámetros actúan igual que el método translate de forma que "e" y "f" indican el desplazamiento en los ejes "x" e "y" respectivamente.

- Los parámetros primero y cuarto actúan igual que el método scale de forma que el parámetro "a" hace un escalado en el eje "x", y el parámetro "d" hace un escalado en el eje "y".

- Los parámetros segundo y tercero actúan como los métodos skew. Aquí el número que debemos indicar no es el número de grados del ángulo sino su tangente. El parámetro "b" inclina el eje "x" un ángulo cuya tangente es la indicada. El parámetro "c" hace lo mismo con el eje "y".

# Transformaciones

- Ejemplo del método matrix:

```
<svg width="270" height="150">  
  <g >  
    <rect x="20" y="25" width="80" height="80"  
fill="orange"/>  
    <text x="30" y="75" style="font: bold 0.8em arial; fill:  
purple;">  
      Original</text>  
  </g>  
  <g transform="matrix(1.2 0.3 -0.9 0.8 190 75)">  
    <rect x="-40" y="-40" width="80" height="80"  
fill="yellow"/>  
    <text x="-30" y="0" style="font: bold 0.8em arial; fill:  
purple;">  
      Matrix</text>  
  </g>  
</svg>
```



# Ejercicio 1

- Realiza una composición, utilizando en la misma al menos un degradado radial, y la inclusión de una imagen desde SVG.

## Ejercicio 2

- Anima la composición realizada, mediante el uso de CSS y JavaScript. Incluye en la animación un cambio en las coordenadas del degradado radial.

# Ejercicio 3 (opcional)

- Utiliza la librería Snap.svg para realizar una animación.