
IIC2026

Ayudantía 1 - HTML, SVG e intro a JS

— Camila Basulto Correa
camila.basulto@uc.cl —

Temas de la ayudantía

1. HTML

- a. ¿Qué es? | ¿Cómo es la estructura?
- b. Tags útiles

2. CSS

- a. ¿Qué es? | Estructura e implementación
- b. Atributos ***class***, ***id***.

3. SVG

- a. ¿Qué es?
- b. Figuras básicas, atributos avanzados.

4. Actividad con SVG

5. Intro a JavaScript

HTML

HTML

HyperText Markup Language

- Lenguaje de marcado o escritura. **NO de programación.**

Ejemplos: Markdown, XML, TeX.

- Permiten agregar enlaces, multimedia e incluso... Visualizaciones :D (aunque solo estáticas.... necesitamos de un lenguaje de programación si queremos agregar interactividad).
- Se maneja mediante **etiquetas o tags.**

HTML - Estructura completa

```
<!DOCTYPE html>
<html>
  <!-- Comentario -->
  <head>
    <meta charset="utf-8">
    <title>Título de documento completo</title>
  </head>

  <body>
    <h1> Un título </h1>
    <p> Un párrafo </p>
    <h2> Un subtítulo </h2>
    <p> Otro párrafo </p>
    <ul>
      <li>Primer elemento de lista.</li>
      <li>Segundo elemento de lista.</li>
      <li>Tercer elemento de lista.</li>
    </ul>

  </body>
</html>
```

1. **Meta** → Definir información del documento en sí. En este caso la codificación (utf-8 recomendado).
2. **Title** → Agregar título al documento (nombre de la ventana).

Un título

Un párrafo

Un subtítulo

Otro párrafo

- Primer elemento de lista.
- Segundo elemento de lista.
- Tercer elemento de lista.

Algunos tags útiles

- `Descripción de la imagen`
- `Lista de tags`
- `<progress value="50" max="100"> 50% </progress>`
- `<label for="saboresForm">Escoja un sabor:</label>`
- `<select name="sabores" id="saboresForm">`
 - `<option value="chocolate" selected>Chocolate</option>`
 - `<option value="vainilla">Vainilla</option>`
 - ...
- `</select>`
- `<title>Nombre de la ventana</title>`
- `
`
- `<div></div>`
- `, <i></i>, <u></u>, <!--comentario-->`

CSS

CSS y class | id

Cascading Style Sheets

- Lenguaje de hojas de estilo.
- Permiten declarar propiedades visuales para HTML. El navegador web es quien interpreta dichas declaraciones.

Internal CSS

`<style>`

```
h1 {  
    margin-left: 50px;  
    color: magenta;  
}
```

`</style>`

Inline CSS

```
<p style="font-size:15px; font-family: 'Times New  
Roman', Arial; color:#C6352C;">Texto</p>
```

External CSS

`<link href="estilo.css" rel="stylesheet">`

```
1  
2  v body{background-color: ■rosybrown;  
3      font-family: monospace;  
4      font-size: 20px;  
5  }  
6  
7  v #identificador{  
8      color: ■red;  
9  }  
10  
11 v .clase1{  
12     font-size:30px;  
13 }  
14
```


SVG

SVG

Scalable **V**ector **G**raphics

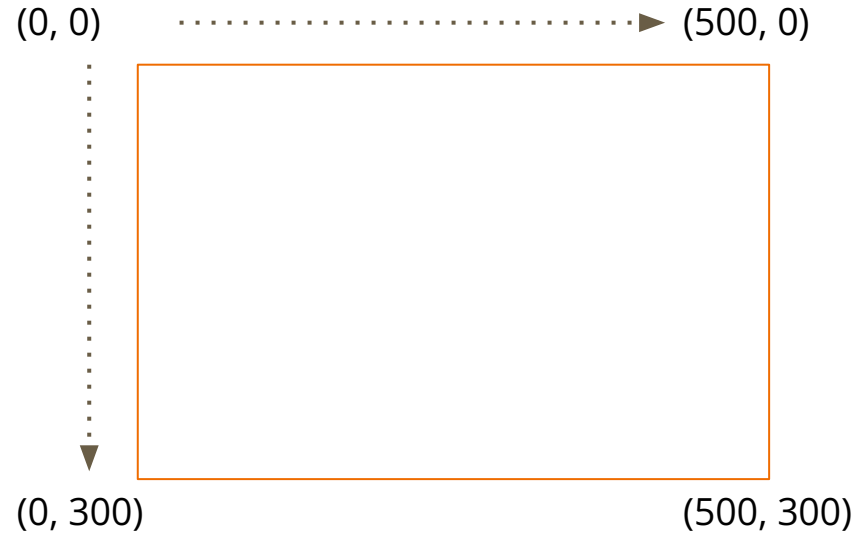
- Formato de imágenes bidimensionales **vectorial**.
 - Como no son pixeles, la calidad es mucho mayor.
- Define su contenido a partir de objetos con propiedades.
- Muy compatible con HTML.
- Los objetos aparecen y se superponen entre ellos según cómo los coloquemos en el código.

SVG - Formato

```
<svg width="500" height="300">
```

...

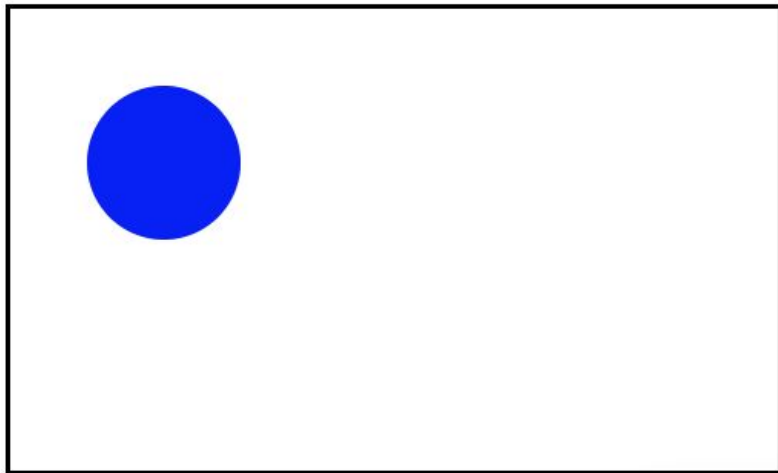
```
</svg>
```



SVG - circle

```
<svg width="500" height="300">  
  <circle cx="100" cy="100" r="50" fill="#0000FF" />  
</svg>
```

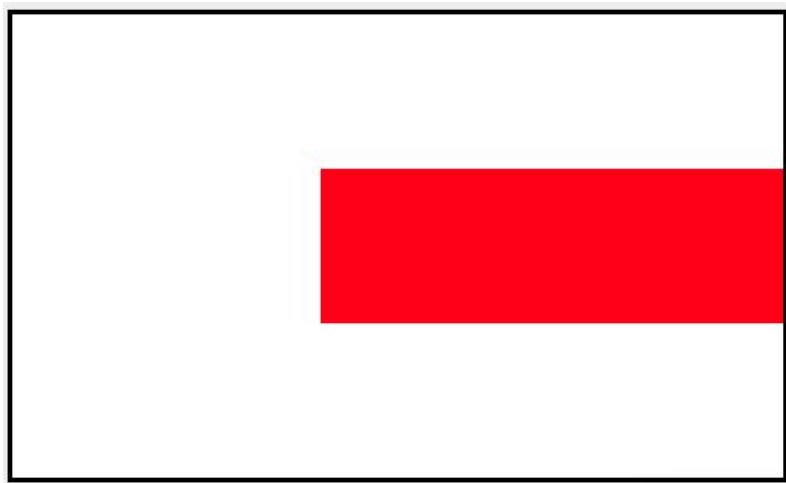
- Tag `<circle/>` para crear un **círculo**.
- `cx="100"` → Indica centro en eje X
- `cy="100"` → Indica centro en eje Y
- `r="50"` → Indica radio del círculo
- `fill="#0000FF"` → Indica color de relleno



SVG - rect

```
<svg width="500" height="300">  
  <rect x="200" y="100" width="300" height="100" fill="red" />  
</svg>
```

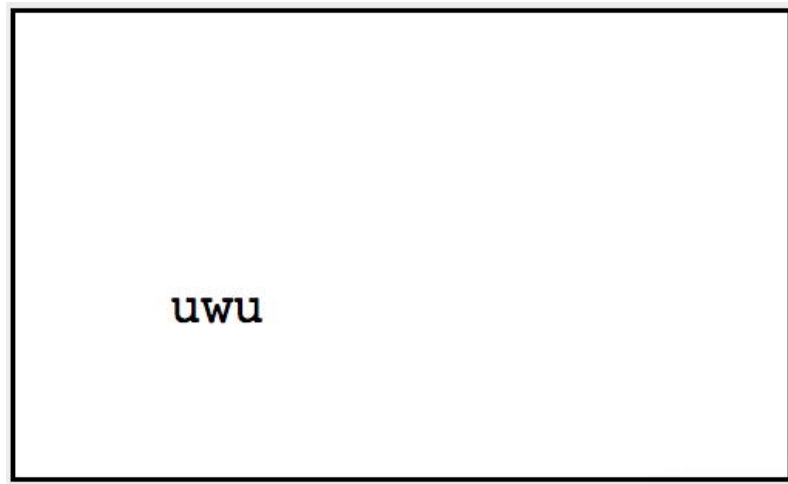
- Tag `<rect/>` para crear un **rectángulo**.
- `x="200"` → Indica donde estará la esquina izquierda superior en el eje X.
- `y="100"` → Indica donde estará la esquina izquierda superior en el eje Y.
- `width="300"` → Indica el ancho.
- `height="100"` → Indica la altura.
- `fill="red"` → Indica color de relleno.



SVG - text

```
<svg width="500" height="300">  
  <text x="100" y="200" font-size="34" size="1">uwu </text>  
</svg>
```

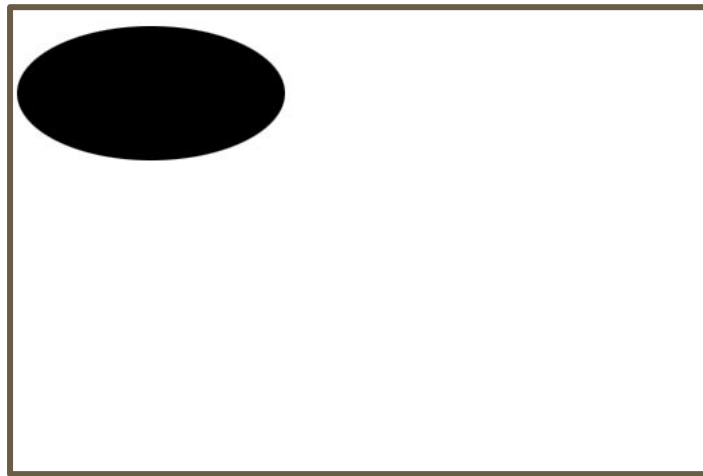
- Tag `<text/>` para crear un **texto**.
- Dentro del tag se escribe el texto a mostrar
- `x="100"` → Indica donde estará la esquina izquierda superior en el eje X.
- `y="200"` → Indica donde estará la base del texto en el eje Y.
- `font-size="34"` → Indica el tamaño



SVG Avanzado - ellipse

```
<svg width="500" height="300">  
  <ellipse cx="100" cy="50" rx="100" ry="50" />  
</svg>
```

- Tag `<ellipse/>` para crear una **Ellipse**.
- `cx="100"` → Indica centro en eje X
- `cy="50"` → Indica centro en eje Y
- `rx="50"` → Indica radio del círculo en eje X
- `ry="100"` → Indica radio del círculo en eje Y
- `fill` indica el color de relleno, si no especificamos es negro por defecto.

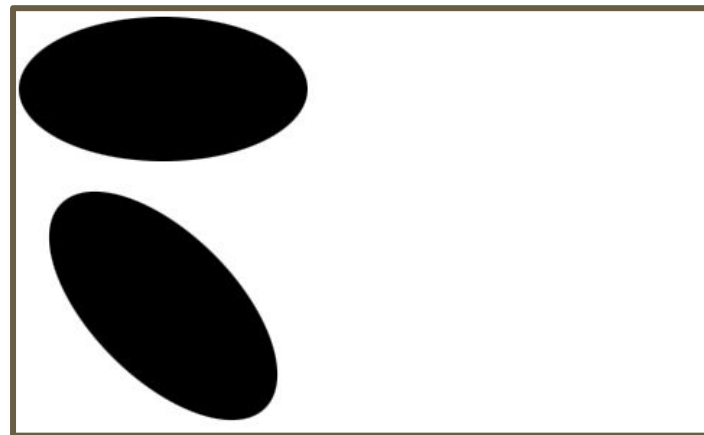


SVG Avanzado - transform

```
<svg width="500" height="300">  
  <ellipse cx="100" cy="50" rx="100" ry="50" />  
  <ellipse cx="100" cy="50" rx="100" ry="50" transform="translate(0, 150) rotate(45 100 50)" />  
</svg>
```

Como su nombre indica, el atributo **transform** permite aplicar transformaciones a las figuras.

- **translate**: permite mover trasladar un elemento.
 - El primer elemento es la traslación en X
 - El segundo elemento es la traslación en Y
- **rotate**: permite rotar un elemento
 - El primer elemento son los grados de rotación
 - El segundo elemento es el punto de origen en X de la rotación.
 - El tercer elemento es el punto de origen en Y de la rotación

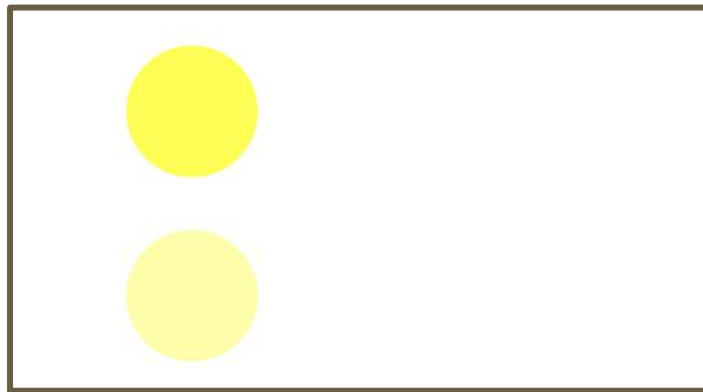


SVG Avanzado - opacity

```
<svg width="500" height="300">  
  <circle cx="150" cy="100" r="50" fill="yellow"/>  
  <circle cx="150" cy="100" r="50" fill="yellow" opacity="0.5"/>  
</svg>
```

Como su nombre indica, el atributo **opacity** permite cambiar la opacidad (transparencia) de las figuras.

- El valor va entre 0 y 1.
- 0 = nada opaco, es decir, 100% transparente.
- 1 = totalmente opaco, es decir, 0% transparente.



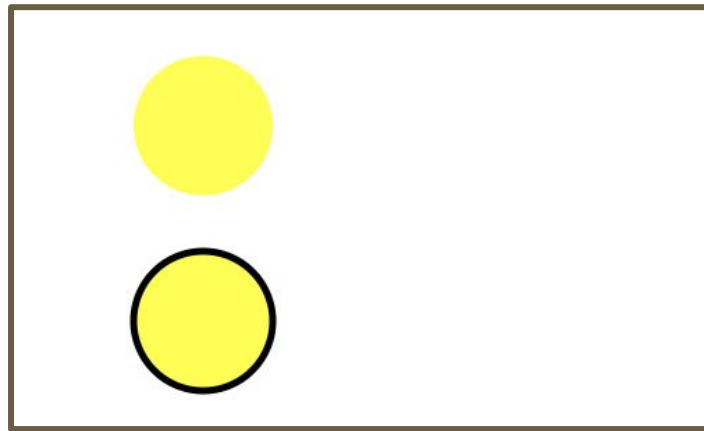
SVG Avanzado - stroke y stroke-width

```
<svg width="500" height="300">
  <circle cx="150" cy="100" r="50" fill="yellow"/>
  <circle cx="150" cy="100" r="50" fill="yellow" stroke="black" stroke-width="5px"/>
</svg>
```

Como sus nombres indican, estos atributos sirven para definir el borde y ancho del borde de las figuras.

- **stroke**: define el color del borde.
- **stroke-width**: define el ancho del borde.

⚠ Si **stroke-width** es 10px en un círculo. El círculo será 5px más grande y el relleno será comido en 5px. Es decir, **el borde crece equitativamente para afuera y para dentro**.



Incluir SVG en el HTML

Tag de SVG

```
1 <svg width="400" height="400" style="border: 1px solid black">
2   <!-- Carita -->
3   <circle cx="200" cy="200" r="100" fill="#FEE12B" />
4
5   <!-- Mejillas -->
6   <circle cx="130" cy="230" r="25" fill="red" />
7   <circle cx="270" cy="230" r="25" fill="red" />
8
9   <!-- Ojos -->
10  <circle cx="150" cy="190" r="25" fill="black" />
11  <circle cx="250" cy="190" r="25" fill="black" />
12  <!-- Brillo -->
13  <circle cx="140" cy="180" r="8" fill="white" />
14  <circle cx="240" cy="180" r="8" fill="white" />
15
16 </svg>
```

SVG como imagen

```
<body>
  </img>
</body>
```

```
1 <svg height="300" width="300" xmlns="http://www.w3.org/2000/svg">
2   <path
3     d="M269.352 91.664h-42.436V80.04h70.577v-65.73h-14.294V0H168.
4     fill="#666" />
5   <path d="M182.248 20.569h16.974V37.56h-16.974z" fill="#fff" />
6 </svg>
7
```

```
<?xml version="1.0" standalone="no"?>
<svg xmlns:xlink="http://www.w3.org/1999/xlink" xmlns="http://www.w3.org/2000/svg"
  <path fill="magenta" d="M 10,30 A 20,20 0,0,1 50,30 A 20,20 0,0,1 90,30 Q 90,6
</svg>
```

NO es el método que utilizarán en el curso

SVG - Ejemplo

```
1 <svg width="400" height="400" style="border: 1px solid black">
2   <!-- Carita -->
3   <circle cx="200" cy="200" r="100" fill="#FEE12B" />
4
5   <!-- Mejillas -->
6   <circle cx="130" cy="230" r="25" fill="red" />
7   <circle cx="270" cy="230" r="25" fill="red" />
8
9   <!-- Ojos -->
10  <circle cx="150" cy="190" r="25" fill="black" />
11  <circle cx="250" cy="190" r="25" fill="black" />
12  <!-- Brillo -->
13  <circle cx="140" cy="180" r="8" fill="white" />
14  <circle cx="240" cy="180" r="8" fill="white" />
15
16  <!-- Nariz -->
17  <circle cx="200" cy="210" r="8" fill="black" />
18  <rect x="190" y="190" width="20" height="20" fill="#FEE12B" />
19
20  <!-- Boca -->
21  <circle cx="185" cy="230" r="15" fill="black" />
22  <circle cx="185" cy="225" r="15" fill="#FEE12B" />
23  <circle cx="212" cy="230" r="15" fill="black" />
24  <circle cx="212" cy="225" r="15" fill="#FEE12B" />
25
26  <!-- Orejas -->
27  <ellipse cx="120" cy="110" rx="30" ry="80" fill="black" transform="rotate(350 120 110)"/>
28  <ellipse cx="280" cy="110" rx="30" ry="80" fill="black" transform="rotate(10 280 110)"/>
29
30  <ellipse cx="130" cy="130" rx="35" ry="80" fill="#FEE12B" transform="rotate(350 120 130)"/>
31  <ellipse cx="270" cy="130" rx="35" ry="80" fill="#FEE12B" transform="rotate(10 270 130)"/>
```



Practica - Bob Esponja

Con SVG intenta recrear la siguiente imagen en un archivo HTML (bob_esponja.html). No tiene que ser exactamente igual, solo sigue el concepto general.



Consejo: utilizar [Color Picker](#) para los códigos de color. La imagen de referencia está subida.

Intro a JavaScript

- En el HTML

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <meta charset="utf-8">
```

```
    <title>Ejemplo con JS</title>
```

```
  </head>
```

```
  <body id="raiz">
```

```
    <h1 id="principal">¡Presioname!</h1>
```

```
    <script src='programa.js' charset='utf-8'></script>
```

```
  </body>
```

```
</html>
```



Programa a
ejecutar

```
// programa.js
```

```
const cuerpo = document.getElementById("cuerpo");  
const principal = document.getElementById("principal");  
  
let contador = 0;  
principal.addEventListener("click", () => {  
  contador += 1;  
  const parrafo = document.createElement("p");  
  const texto = document.createTextNode(`Cantidad de clics o.o:  
${contador}`);  
  parrafo.appendChild(texto);  
  
  cuerpo.appendChild(parrafo);  
});
```

IIC2026

Ayudantía 1 - HTML y SVG

Camila Basulto Correa
camila.basulto@uc.cl
