

---

# IIC2026

## Visualización de Información

— Hernán F. Valdivieso López —  
(2024 - 1 / Clase 02)

---

# Temas de la clase - HTML, CSS y SVG

1. Introducción al desarrollo Web.
  - a. HTML
  - b. CSS
2. Introducción a SVG.

# Introducción a desarrollo web (HTML y CSS)

---

# Introducción al desarrollo Web

¿Qué tiene que ver esto con la visualización? 🤔

1. La web es uno de los medios de comunicación más importantes.
  - a. Redes sociales
  - b. Noticias
  - c. Entre otros
2. Es transitada de forma masiva todos los días.
3. La web permite formas de interacción que sobrepasan otros medios.
  - a. Videos/Gifs.
  - b. Interacciones como usar el mouse o el teclado.

**¡La web es un lugar perfecto para incluir nuestras visualizaciones!**

# Introducción al desarrollo Web

¡La web es un lugar perfecto para incluir nuestras visualizaciones!

Para lograr esto, tendremos que aprender varias cosas:

1. **HTML** para la estructura de nuestras páginas.
2. **CSS** para el estilo de nuestras páginas.
3. **SVG** para entender cómo “dibujar” una visualización.
4. **Javascript (JS)** para agregar el componente lógico en nuestras páginas (clicks, abrir datasets, crear la visualización en sí, scroll, etc.).
5. **D3.js** para crear las visualizaciones de forma “más amigable” y totalmente personalizables.

# HTML

## HyperText Markup Language

- Lenguaje de marcado o escritura. **NO de programación.**
- Una de las tecnologías básicas de la web. Toda página web en funcionamiento es un documento HTML.
- Un documento HTML es un documento de texto con información añadida.
- Permiten agregar enlaces, multimedia e incluso... Visualizaciones :D (aunque solo estáticas.... necesitamos de un lenguaje de programación si queremos agregar interactividad).
- Es el trabajo de un navegador web leer estos archivos de texto e interpretarlos visualmente.

# HTML - Formato

¿Cómo funciona?

- A partir de `<etiquetas> Contenido </etiquetas>`
- Si no tienen contenido, la etiqueta **puede ser** `<etiquetas />`

Por ejemplo:

- `<h1> Título </h1>`
- `<h2> Subtítulo </h2>`
- `<p> Parrafo </p>`
- ``
- ` </img>`

# HTML - Tags

`<h1>` Un título `</h1>`

`<p>` Un párrafo `</p>`

`<h2>` Un subtítulo `</h2>`

`<p>` Otro párrafo `</p>`

**Un título**

Un párrafo

**Un subtítulo**

Otro párrafo



# HTML - Tags

```
<ul>  
  <li>Primer elemento de lista.</li>  
  <li>Segundo elemento de lista.</li>  
  <li>Tercer elemento de lista.</li>  
</ul>
```

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

# HTML - Estructura completa

```
<!DOCTYPE html>
```

Línea inicial para indicar el tipo de documento

```
<html>
```

Tag para indicar el inicio del HTML

```
<!-- Comentario -->
```

**Opcional:** Forma de agregar comentarios

```
<head>
```

Tag para contenido **no visual** (metadata, CSS inline, import a código CSS y JS, etc.)

```
</head>
```

```
<body>
```

Tag para contenido **visual** (títulos, visualizaciones, vídeos, imágenes, etc.)

```
</body>
```

```
</html>
```

# HTML - Estructura completa

```
<!DOCTYPE html>
<html>
  <!-- Comentario -->
  <head>
  </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>
```

# Un título

Un párrafo

## Un subtítulo

Otro párrafo

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

# HTML - Estructura completa

```
<!DOCTYPE html>
<html>
  <!-- Comentario -->
  <head>
    <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>
```

Agregar información **no visual**:

**Title** → Agregar título al documento. Esto se verá en la pestaña del navegador.

Puedes poner muchas más cosas en el *head*: [HTML head Elements](#)

# HTML - Atributos

```
<a href="https://www.crunchyroll.com/">  
    Soy un enlace a un sitio para ver anime!  
</a>
```

[Soy un enlace a un sitio para ver anime!](https://www.crunchyroll.com/)

El tag `<a>` es para definir enlaces.

- Usamos el atributo `href` para indicar la URL del enlace. Entregamos un string que será la URL del enlace.
- El contenido dentro de la etiqueta es el texto que tendrá el hipervínculo.

# HTML - Atributos

```

```

El tag `<img>` es para agregar imágenes.

- Usamos el atributo `width` para definir el tamaño. Usamos `"200px"` para indicar el tamaño.
- Usamos el atributo `src` para definir la ruta a la imagen. Puede ser un string a una página web o a un archivo.

# HTML - Atributos globales (id y class)

## id

```
<h1 id="principal"> Soy el único título principal </h1>
```

```
<h1> Yo soy un título pero no principal :c </h1>
```

- Permite identificar **elementos específicos** en todo el documento.
- Es **único**, sólo un elemento puede tener un identificador específico.

## class

```
<p class="importante"> Soy un párrafo importante</p>
```

```
<p class="importante"> También soy importante</p>
```

```
<p> Yo no X.X </p>
```

- Permite identificar **grupos de elementos** en todo el documento.
- **Múltiples elementos** pueden **compartir** un valor de **clase**.

# CSS

## Cascading **S**tyle **S**heets

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



Solo HTML



+CSS



gif\*

+ Javascript (Spoiler)



# CSS - Ejemplo

```
p {  
  color: blue;  
}
```

Esta declaración busca todos los elementos <p> del HTML y les cambia su color a azul

# CSS - Formato general

```
selector {  
  propiedad_1: valor_1;  
  propiedad_2: valor_2;  
  ...  
}
```

A medida que avancemos en el curso, aprenderemos sobre diferentes propiedades y sus posibles valores :D

# CSS - Selectores

```
p {  
  color: blue;  
}
```

Todos los elementos **<p>** cambian su color a azul.

```
li, h1, h2 {  
  color: blue;  
}
```

Todos los elementos **<li>**, **<h1>** y **<h2>** cambian su color a azul.

```
.importante {  
  color: red;  
}
```

Todos los elementos cuya **class="importante"** cambian su color a rojo.

```
#principal {  
  color: green;  
}
```

Todos los elementos cuya **id="principal"** cambian su color a verde.

```
body {  
  background-color: #eeeeee;  
  font-family: monospace;  
  font-size: 20px;  
}
```

El color de fondo del HTML será #eeeeee.

El tipo de letra será mono-espaciado

El tamaño de la letra será de 20 pixeles.

# CSS - ¿Cómo lo incluimos en un HTML?

## Forma 1 - Internal CSS

```
<head>
```

```
<!-- Usamos tag <style> para tener una sección para CSS en el mismo HTML.
```

```
    Esto se llama "Internal CSS". No recomendamos mucho esto porque no  
    nos permite separar bien lo que es HTML y CSS en diferentes archivos
```

```
-->
```

```
<style>
```

```
h1 {
```

```
    margin-left: 50px;
```

```
    color: magenta;
```

```
}
```

```
</style>
```

```
</head>
```

# CSS - ¿Cómo lo incluimos en un HTML?

## Forma 2 - Inline CSS

```
<p style="text-decoration: underline; color: black">
```

Este otro párrafo tiene CSS en el mismo elemento HTML para estar subrayado y un color ^.^.

Esto se llama "Inline CSS" y tampoco lo recomendamos porque es lo mismo que el "Internal CSS"</p>

```
<p>
```

# CSS - ¿Cómo lo incluimos en un HTML?

## Forma 3 - external CSS

```
<head>
  <link href="estilo_ejemplo_4.css"
        rel="stylesheet" type="text/css">
</head>
```

```
/*
estilo_ejemplo_4.css
(estó es un comentario en CSS)
*/

p {
  color: ■blue;

body {
  background-color: □#eeeeee;
  font-family: monospace;
  font-size: 20px;
}
```

# CSS - Muchas formas de personalización

Vimos 3 formas de personalizar:

- *internal css* (tag style)
- *inline css* (atributo style)
- *external css* (archivo .css)

Además, dentro del css hay diferentes selectores →

🤔 ¿Quién gana cuando hay mucho estilo aplicándose? 🤔

Veremos con más detalle esto en la próxima clase 😊

Aquí un [spoiler](#) para los que quieran saber ya 😊

```
p {  
    color: blue;  
}  
  
li, h1, h2 {  
    color: blue;  
}  
  
.importante {  
    color: red;  
}  
  
#principal {  
    color: green;  
}  
  
body {  
    background-color: #eeeeee;  
    font-family: monospace;  
    font-size: 20px;  
}
```

# Ejemplo HTML + CSS (Ejemplo\_2.html y estilo\_ejemplo\_2.css)

```
<!DOCTYPE html>
<html>

<head>
  <title>Título de documento completo</title>
  <link href="estilo_ejemplo_2.css" rel="stylesheet" type="text/css">
</head>

<body>
  <h1 id="principal"> Soy el único título principal </h1>
  <h1> Yo soy un título pero no principal :c </h1>
  <p class="importante"> Soy un párrafo importante</p>
  <h2> Soy subtítulo y además no soy principal uwu</h2>
  <p class="importante"> También soy importante</p>
  <p class="importante"> Yo no X.X</p>

  <ul>
    <li>Primer elemento de lista.</li>
    <li>Segundo elemento de lista.</li>
    <li>Tercer elemento de lista.</li>
  </ul>

  No soy importante U.U,
  <a href="https://www.crunchyroll.com/">
    Soy un enlace a un sitio para ver anime!
  </a>
  </p>
</body>

</html>
```

```
/* estilo.css (esto es un comentario en CSS) */

p {
  color: blue;
}

li, h1, h2 {
  color: blue;
}

.importante {
  color: red;
}

#principal {
  color: green;
}

body {
  background-color: #eeeeee;
  font-family: monospace;
  font-size: 20px;
}
```



## Ejemplo HTML + CSS (Ejemplo\_2.html y estilo\_ejemplo\_2.css)

```
<!DOCTYPE html>
<html>

<head>
    <title>Título de documento completo</title>
    <link href="estilo_ejemplo_2.css" rel="stylesheet" type="text/css">
</head>

<body>
    <h1 id="principal"> Soy el único título principal </h1>
    <h1> Yo soy un título pero no principal :c </h1>
    <p class="importante"> Soy un párrafo importante</p>
    <h2> Soy subtítulo y además no soy principal uwu</h2>
    <p class="importante"> También soy importante</p>
    <p class="importante"> Yo no X.X</p>

    <ul>
        <li>Primer elemento de lista.</li>
        <li>Segundo elemento de lista.</li>
        <li>Tercer elemento de lista.</li>
    </ul>

    
    <p>No soy importante U.U,
        <a href="https://www.crunchyroll.com/">
            Soy un enlace a un sitio para ver anime!
        </a>
    </p>
</body>

</html>
```

## Soy el único título principal

Yo soy un título pero no principal :c

Soy un párrafo importante

Soy subtítulo y además no soy principal uwu

También soy importante

Yo no X.X

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



No soy importante U.U, Soy un enlace a un sitio para ver

```
/* estilo.css (esto es un comentario en CSS) */
```

```
p {
  color: blue;
}
```

```
li, h1, h2 {
  color: blue;
}
```

```
.importante {
  color: red;
}
```

```
#principal {  
  color: green;  
}
```

```
body {
  background-color: #eeeeee;
  font-family: monospace;
  font-size: 20px;
}
```

# Ejemplo HTML + CSS (Ejemplo\_2.html y estilo\_ejemplo\_2.css)

## Mini desafío para la casa

1. Intente cambiar el color de fondo.
2. Intente cambiar el tamaño de letras de los `<p>` a 40 pixeles
3. Intente cambiar el color de texto de todo elemento de clase "importante".
4. Cambie la imagen por la siguiente:

<https://www.crunchyroll.com/imgsrv/display/thumbnail/1200x675/catalog/crunchyroll/2583f9f13ed1f26108ae4279f5b50c94.jpe>

# Introducción a SVG

---

# SVG

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

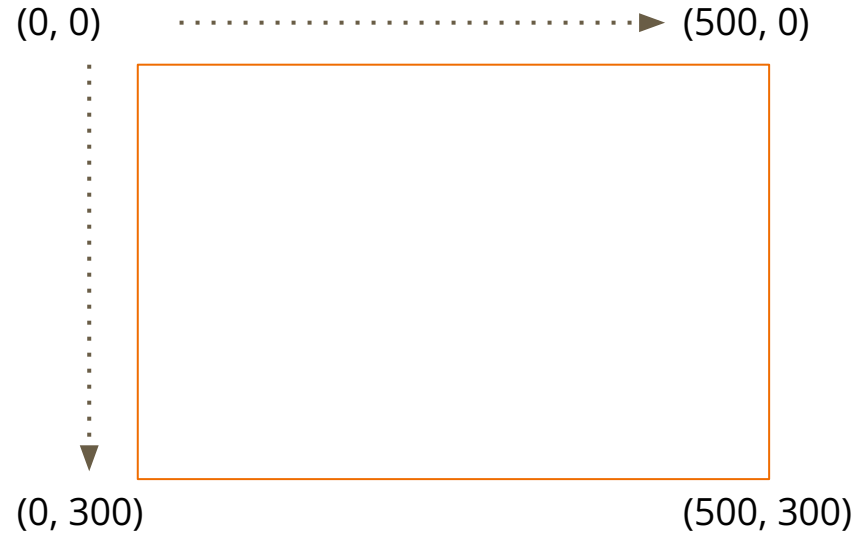
- Formato de imágenes bidimensionales **vectorial**.
  - Como no son pixeles, la calidad es mucho mayor.
- Define su contenido a partir de objetos con propiedades.
- Muy compatible con HTML.

# SVG - Formato

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

...

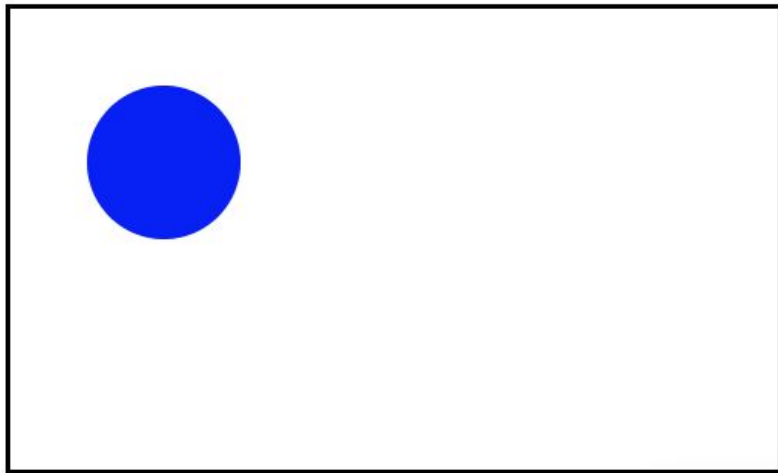
```
</svg>
```



# SVG - Ejemplo

```
<svg width="500" height="300">  
  <circle cx="100" cy="100" r="50" fill="#0000FF" color="red" radius="10000" />  
</svg>
```

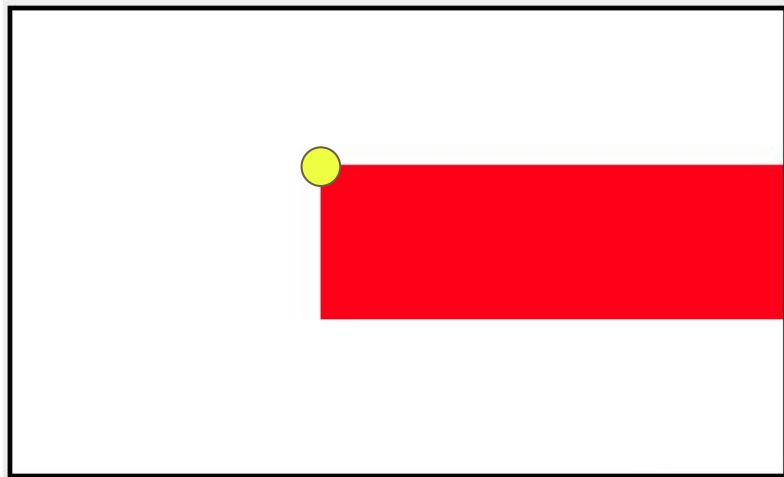
- Tag `<circle/>` para crear un **círculo**.
- `cx="100"` → Indica el centro en el eje X.
- `cy="100"` → Indica el centro en el eje Y.
- `r="50"` → Indica radio del círculo.
- `fill="#0000FF"` → Indica color de relleno (azul).
- `color="magenta"` → **No existe**.
- `radius="10000"` → **No existe**.



# SVG - Ejemplo

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

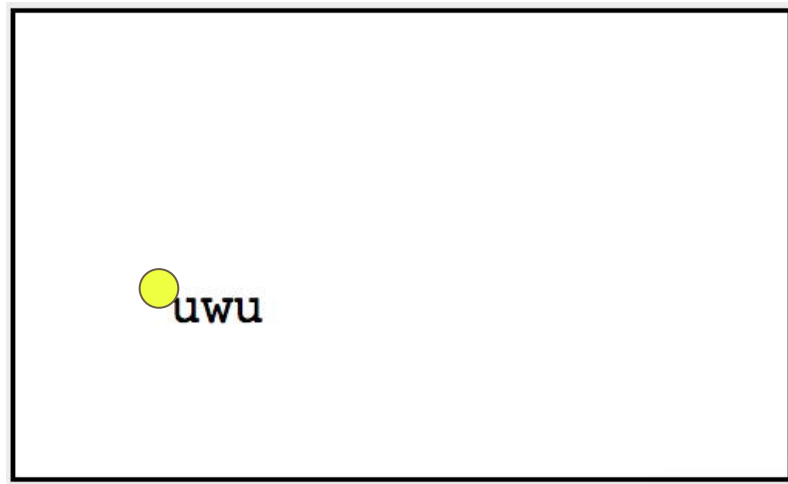
- Tag `<rect/>` para crear un **rectángulo**.
- `x="200"` → Indica donde estará la esquina izquierda superior en el eje X (■).
- `y="100"` → Indica donde estará la esquina izquierda superior en el eje Y (■).
- `width="300"` → Indica el ancho.
- `height="100"` → Indica la altura.
- `fill="red"` → Indica color de relleno.
- `color="magenta"` → **No existe**.



# SVG - Ejemplo

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

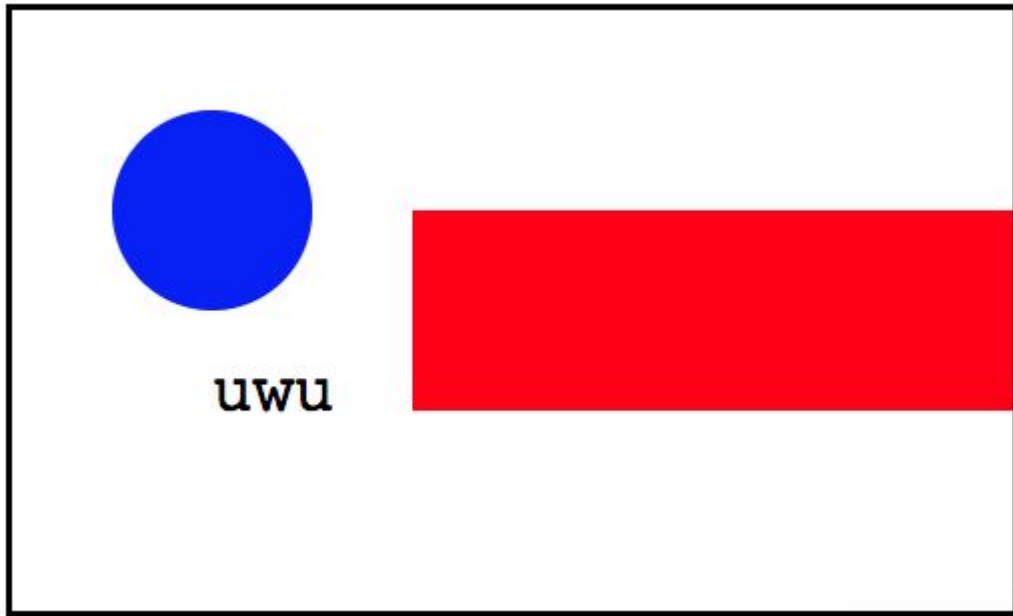
- Tag `<text/>` para crear un **texto**.
- Dentro del tag se pone el texto a mostrar.
- `x="100"` → Indica donde estará la esquina izquierda superior en el eje X (■).
- `y="200"` → Indica donde estará la esquina izquierda superior en el eje Y (■).
- `font-size="34"` → Indica el tamaño del texto.
- `size="1"` → **No existe**.





# SVG - Ejemplo

```
<svg width="500" height="300">  
  <rect x="200" y="100" width="300" height="100" fill="red" color="magenta" />  
  <circle cx="100" cy="100" r="50" fill="#0000FF" color="red" radius="10000" />  
  <text x="100" y="200" font-size="34" size="1">uwu </text>  
</svg>
```



# SVG - Ejemplo

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

```
  <!-- Ojo, "color" no es un atributo de rect, así que no hará nada -->
```

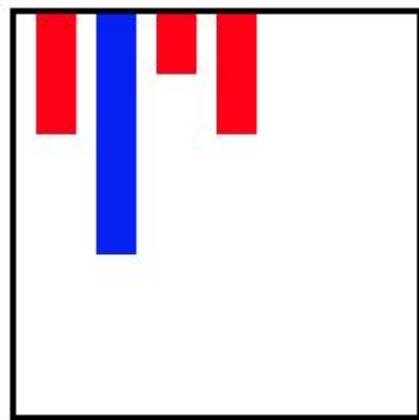
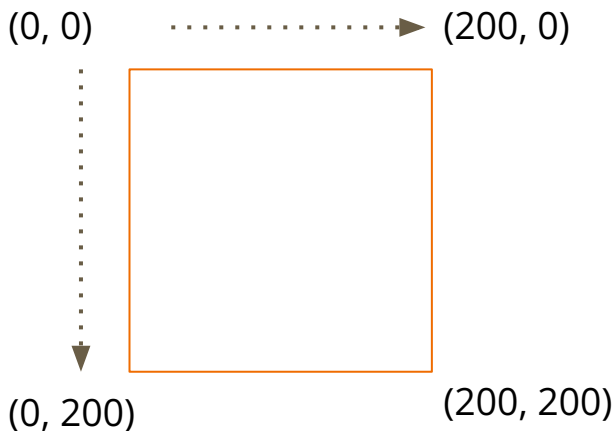
```
  <rect x="10" y="0" width="20" height="60" fill="red" color="magenta"></rect>
```

```
  <rect x="40" y="0" width="20" height="120" fill="blue"></rect>
```

```
  <rect x="70" y="0" width="20" height="30" fill="red"></rect>
```

```
  <rect x="100" y="0" width="20" height="60" fill="red"></rect>
```

```
</svg>
```



# SVG

## Forma 1 - SVG en el HTML

```
<body>
```

```
  <h1> Haremos un SVG en este ejemplo</h1>
```

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

```
    <!-- Ojo, "color" no es un atributo de rect, así que no hará nada -->
```

```
    <rect x="10" y="0" width="20" height="60" fill="red" color="magenta"></rect>
```

```
    <rect x="40" y="0" width="20" height="120" fill="blue"></rect>
```

```
    <rect x="70" y="0" width="20" height="30" fill="red"></rect>
```

```
    <rect x="100" y="0" width="20" height="60" fill="red"></rect>
```

```
  </svg>
```

```
</body>
```

# SVG

Forma 2 - SVG en un archivo y lo incluimos como imagen

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



The screenshot shows a code editor with two tabs: 'corazon.svg' and 'dinosaurio.svg'. The 'corazon.svg' tab is active, displaying the following XML code:

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



The screenshot shows a code editor with two tabs: 'corazon.svg' and 'dinosaurio.svg'. The 'dinosaurio.svg' tab is active, displaying the following XML code:

```
1 <svg height="300" width="300" xmlns="http://www.w3.
2   <path
3     d="M269.352 91.664h-42.436V80.04h70.577v-65
4     fill="#666" />
5   <path d="M182.248 20.569h16.974V37.56h-16.974z"
6 </svg>
```

# Ejemplo Integrador (Ejemplo\_4.html y estilo\_ejemplo\_4.css)

Ejemplo\_4.html es muy grande para ver en pantalla.

Vamos mejor al código 🤗👩👨

```
/*
estilo_ejemplo_4.css
(estó es un comentario en CSS)
*/

p {
    color: blue;
}

body {
    background-color: #eeeeee;
    font-family: monospace;
    font-size: 20px;
}
```

# Ejemplo Integrador (Ejemplo\_4.html y estilo\_ejemplo\_4.css)

## Ejemplo integrando todo

Este es un párrafo :D

Este otro párrafo tiene CSS en el mismo elemento HTML para estar subrayado y un color ^.^. Esto se llama "inline CSS" y tampoco lo recomendamos porque es lo mismo que el "Internal CSS"

Para saber más de CSS, visitar este [link](#)



uwu



# Antes de salir...

Se publicó un mini control de alternativas en Canvas sobre lo que **vimos en la clase de hoy**. El mini control lo puedes encontrar en la sección "Módulos" de Canvas.

- **Duración:** 2 semanas para realizarlo a partir de hoy. Una vez terminado el plazo, tendrán retroalimentación por cada pregunta.
- **Intentos para responder:** ilimitados.
- **Extensión:** 6 preguntas de 1 punto c/u + 1 pregunta de 2 puntos.
- **Condición para obtener el punto RC:** Al menos 7 puntos de 8.
- Cada vez que respondan, verán el puntaje total logrado, pero no cuáles preguntas están correctas e incorrectas.

# Próximos eventos

## Próxima clase (jueves)

- Más código → Más cosas de SVG e introducción a Javascript.

## Ayudantía del viernes

- HTML y SVG básico + Intro a JS
- Para quienes tienen únicamente IIC1103 se recomienda fuertemente asistir.



---

# IIC2026

## Visualización de Información

— Hernán F. Valdivieso López —  
(2024 - 1 / Clase 02)

---