



# INTRODUCCIÓN A CSS

LEONARDO CHAGOYA  
KAREN PÉREZ



# HTML Y CSS ADICIONAL

```
<section class="buttons">
```

Es un contenedor semántico: agrupa elementos que tienen relación entre sí (en este caso, dos botones).

La clase "buttons" es un nombre que tú eliges, sirve para aplicar estilos CSS a ese bloque.

No cambia el comportamiento, solo permite identificar esa sección fácilmente.



# BUTTON Y SPAN

```
<section class="buttons">  <!-- Se abre una sección HTML que agrupa los botones relacionados -->
    <button class="btn play"><span>PLAY</span></button>
        <!-- <class="btn play": este botón tiene dos clases uno define un estilo general para todos los botones y la otra específicamente a PLAY-->
    <button class="btn list">
        <span>MY LIST</span> <!-- se usa un <span> para envolver el texto.
            | | | | | | | | Esto permite estilizar el texto dentro del botón por separado (por ejemplo, cambiar el color, animarlo, etc.) -->
        <svg><!--Define un gráfico vectorial. Se usa mucho para efectos visuales, como animaciones al hacer hover. -->
            <rect x="0" y="0" fill="none"></rect> <!-- Es una etiqueta de SVG que dibuja un rectángulo. Para que funcione bien, se le deben definir atributos-->
        </svg>
    </button>
</section>
```

## VIDEO

Se crea un contenedor general para el modal de video (una ventana emergente).

- **id="videoModal"**: permite identificar este bloque con JavaScript (para abrir/cerrar el modal)
- **class="modal"**: se usa para aplicar estilos visuales (como fondo oscuro, posición centrada, etc.).

## VIDEO

```
<div class="modal-content">
```

Se crea un contenedor general para el modal de video (una ventana emergente).

**id="videoModal"**: permite identificar este bloque con JavaScript (para abrir/cerrar el modal).

**class="modal"**: se usa para aplicar estilos visuales (como fondo oscuro, posición centrada, etc.).

```
<div class="modal-content">
```

Esta es la caja interior del modal, donde va el contenido que se muestra al usuario.

## FONTS

Font Awesome es una biblioteca de íconos que puedes usar fácilmente en tus páginas web. Permite insertar íconos vectoriales escalables con solo unas líneas de código, sin necesidad de usar imágenes externas.

### ¿Cómo se usa Font Awesome?

Paso 1: Agrega el enlace a tu HTML

Puedes usar su CDN (Content Delivery Network) para no instalar nada:

```
<head>
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.5.0/css/all.min.css">
</head>
```

Paso 2: Usa los íconos con clases en tus etiquetas

```
<i class="fas fa-camera"></i> Cámara
<i class="fab fa-facebook"></i> Facebook
<i class="fas fa-check-circle"></i> Aprobado
```

## ¿QUE PROCEDE?

Ahora que sabemos estructurar una pagina de HTML, seguramente te preguntas:

- ¿Cómo hago mi texto rojo o negro?
- ¿Cómo hago que mi contenido se muestre en tal y tal lugar de la pantalla?
- ¿Cómo decoro mi página web con imágenes de fondo y colores?



# CSS BÁSICO

- CSS (Cascading Style Sheets/Hojas de Estilo en Cascada)
- Es el código que usas para dar estilo a tu página web. CSS Básico te lleva a través de lo que tú necesitas para empezar.
- Permite separar el contenido (HTML) de la presentación.





## Ventajas



Separación de contenido y presentación



Facilita la gestión y mantenimiento de los sitios web.

Mayor eficiencia y ahorro de tiempo



CSS permite reutilizar estilos en múltiples páginas web

Diseño responsive y adaptabilidad



CSS asegura que el diseño de un sitio web sea consistente en todas sus páginas

Consistencia en el diseño



Funcionalidades y bases de datos

Compatibilidad con estándares web



CSS es un estándar web compatible con la mayoría de los navegadores.



## Desventajas



Complejidad y curva de aprendizaje



Debido a su sintaxis y conceptos como especificidad y herencia

Rendimiento



Un uso excesivo o ineficiente de CSS puede ralentizar el tiempo de carga de las páginas web

Problemas de seguridad



Si no se utiliza correctamente, CSS puede ser vulnerable a ciertos ataques de seguridad.

Depuración



Depurar CSS, especialmente con preprocesadores, puede ser complicado y requerir herramientas adicionales.

Compatibilidad con estándares web



CSS es un estándar web compatible con la mayoría de los navegadores.

# ¿Cómo se integran HTML y CSS?

CSS puede usarse de tres formas principales con HTML

1. Estilos en línea (inline): Se colocan directamente en la etiqueta HTML usando el atributo **style**.

```
<p style="color: red;">Hola, mundo</p>
```

2. Estilos internos (en el **<head>**):  
Se escriben dentro de una  
etiqueta **<style>** en el mismo  
archivo HTML.

```
<head>
  <style>
    p {
      color: red;
    }
  </style>
</head>
```

3. Estilos externos (archivo .css):  
Se enlaza un **archivo externo**  
usando **<link>**. Esta es la mejor  
práctica.

```
<head>
  <link rel="stylesheet" href="estilos.css">
</head>
</head>
```

# ¿Cómo funciona CSS?

CSS trabaja usando reglas de estilo que se aplican a elementos HTML.  
Cada regla tiene un selector y un bloque de declaraciones.

```
p {  
    color: red;  
    font-size: 18px;  
}
```

- Selector: **p** → indica a qué elementos HTML se aplica (en este caso, a todos los **<p>**).
- Propiedades y valores: lo que se va a modificar (color, tamaño, bordes, etc.).

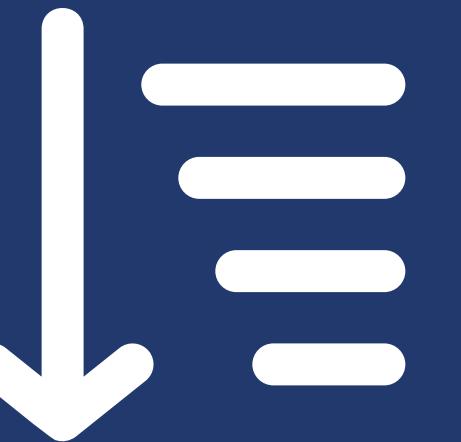
# ¿Qué puedes hacer con CSS?

Colores y fuentes



Times New Roman

Espaciado



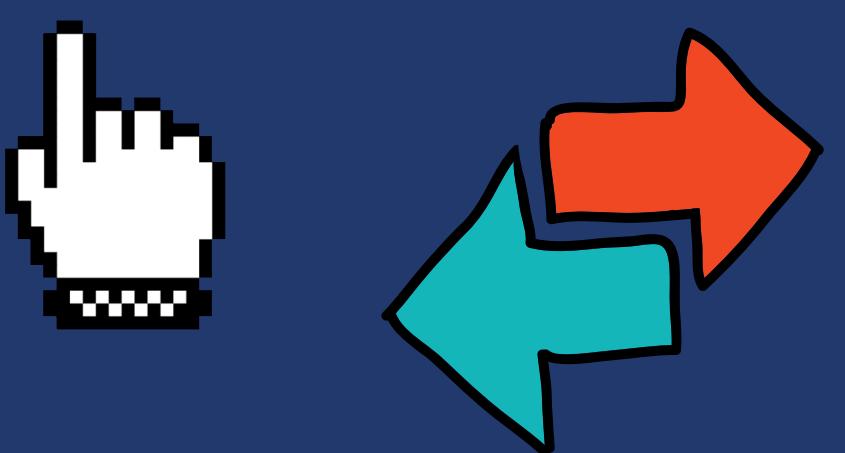
Tamaños y bordes



Posición



Animaciones y transiciones



Diseño responsivo



# ¿Por qué se llama "en cascada"?

Porque cuando hay varias reglas aplicando al mismo elemento, el navegador decide cuál aplicar con base en:

- Especificidad (ID > clase > etiqueta)
- Orden (la última regla sobrescribe a las anteriores si tienen igual peso)
- !important (fuerza una regla a ser aplicada sobre todas)

códigofacilito	!important	1, 0, 0, 0, 0
	#ID	0, 1, 0, 0, 0
	Estilos en línea	0, 0, 1, 0, 0
	.Clases (pseudoclases o atributos)	0, 0, 0, 1, 0
	<Elementos/> (o pseudo elementos)	0, 0, 0, 0, 1
	*Selector Universal	0, 0, 0, 0, 0