

# Introducción a CSS



Fundamentos de CSS

Ing. Mauricio Alejandro Quezada Bustillo

The background consists of a collage of overlapping, semi-transparent paper squares in various colors including shades of purple, teal, orange, grey, and green. Scattered across these squares are several speech bubbles of different colors (yellow, blue, green, dark blue, maroon) and shapes, some pointing towards the center. The overall effect is a layered, artistic composition.

# ¿Qué es CSS?

CSS es el lenguaje que se utiliza para **dar estilo y presentación** a una página web creada con HTML.

# ¿Para qué sirve CSS?



Cambiar colores



Modificar tamaños de  
texto



Ajustar márgenes y  
espacios



Mejorar la apariencia  
visual de un sitio web




# HTML vs CSS




## HTML


- Define la **estructura** del contenido
- Indica *qué es cada cosa*:
  - Títulos
  - Párrafos
  - Listas
  - Imágenes
- 📌 HTML responde a: **¿qué contenido hay?**

## CSS

- Define la **apariciencia visual**
  - Controla *cómo se ve*:
    - Colores
    - Tamaños
    - Espaciados
    - Tipografía
  - 📌 CSS responde a: **¿cómo se ve el contenido?**
- 




# ¿Qué pasa si no usamos CSS?




## Sin CSS

- Texto negro por defecto
- Todo alineado a la izquierda
- Sin espacios ni jerarquía visual
- Difícil de leer y poco atractiva
- 📌 Así se vería una web **solo con HTML**.



## Con CSS


- Colores definidos
  - Tamaños de texto diferenciados
  - Espacios entre elementos
  - Apariencia más clara y profesional
  - 📌 La información es la misma,  
👉 **la experiencia visual cambia totalmente.**
- 




# Formas de aplicar CSS

## Existen 3 formas de aplicar CSS:

- CSS en línea (inline)
  - CSS interno
  - CSS externo
  - 📌 Todas funcionan, pero **no todas se recomiendan igual.**
- 
- 



# CSS en línea (Inline)




Se escribe **directamente en la etiqueta HTML**.

```
<p style="color: red; font-size: 18px;">
```

Texto con estilo en línea


```
</p>
```

- ✓ Fácil y rápido
  - ✗ Difícil de mantener
  - ✗ No se recomienda para proyectos reales
- 

# CSS Interno

```
<!DOCTYPE html>
<html>
<head>
  <style>
    p {
      color: blue;
      font-size: 16px;
    }
  </style>
</head>
<body>
  <p>Texto con CSS interno</p>
</body>
</html>
```

## ¿Qué es CSS interno?

- El CSS se escribe dentro del archivo HTML, usando la etiqueta <style>.
-  Normalmente va dentro de <head>.

### • Ventajas

- ✓ Más ordenado que inline
- ✓ Útil para ejemplos pequeños o prácticas

### • Desventajas

- ✗ El estilo solo sirve para ese archivo
- ✗ No es escalable para proyectos grandes



# CSS Externo

## ¿Qué es CSS externo?

- El CSS se escribe en un archivo **.css separado** y se conecta al HTML.

## Ventajas

- ✓ Separación de estructura y estilo
- ✓ Reutilizable en varias páginas
- ✓ Más orden y mantenimiento
- ✓ Estándar profesional

```
p {  
  color: green;  
  font-size: 16px;  
}
```

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




# Sintaxis básica de CSS



## Estructura de una regla CSS

```
selector {  
  propiedad: valor;  
}
```

## Partes de una regla


- **Selector** → indica *qué elemento* se va a modificar
  - **Propiedad** → indica *qué característica* se cambia
  - **Valor** → define *cómo se cambia*
- 



# Sintaxis básica de CSS



## Reglas importantes

- Cada propiedad termina en ;
  - Las llaves { } agrupan los estilos
  - Se pueden usar varias propiedades en un mismo selector
- 

# Selectores básicos

## Selector por etiqueta

Aplica estilo a **todas las etiquetas del mismo tipo**.

```
p{  
  color: black;  
}
```

 Afecta a **todos los <p>** del documento.



# Selectores básicos



## Selector por clase

Aplica estilo solo a los elementos que tengan esa clase.

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

```
<p class="texto-importante">Texto  
destacado</p>
```





# Selector por ID



Aplica estilo a **un solo elemento único**.

```
#titulo-principal {  
  color: blue;  
}
```

```
<h1 id="titulo-principal">Título  
principal</h1>
```

 Un id **no debe repetirse**.





# Selector universal

Aplica estilo a **todos los elementos**.

```
* {  
  margin: 0;  
}
```

 Se usa con cuidado.





# Colores en CSS


## Cambiar el color del texto

### Propiedad color

Define el color del texto.

```
p{  
  color: red;  
}
```





# Formas de definir colores

❶ Nombres de colores


color: blue;

❷ Código hexadecimal

color: #ff0000;

❸ RGB

color: rgb(0, 128, 0);





# Fondos en CSS

## Propiedad background-color

Define el color de fondo de un elemento.

```
div {  
  background-color: lightgray;  
}
```



# Tipografía en CSS



## Propiedad font-family

Define el tipo de letra.

```
body {  
  font-family: Arial, sans-serif;  
}
```

 Usa fuentes del sistema por ahora.





# Tipografía en CSS

## **Propiedad font-size**

Cambia el tamaño del texto.

```
p{  
  font-size: 16px;  
}
```




# Tipografía en CSS


## **Propiedad font-weight**

Define el grosor del texto.

```
h1 {  
  font-weight: bold;  
}
```



# Alineación y decoración del texto





## Propiedad text-align

Alinea el texto horizontalmente.


```
h1 {  
  text-align: center;  
}
```

Valores comunes:

- left
  - center
  - right
  - justify
- 




# Alineación y decoración del texto



## Propiedad text-decoration

Agrega o quita decoraciones.

```
a{  
  text-decoration: none;  
}
```

- Otros valores:
  - underline
  - line-through
- 



# Introducción al Box Model




## ¿Qué es el Box Model?

En CSS, **todo elemento es una caja**.

Cada caja está compuesta por:

- Contenido
- Padding
- Border
- Margin

## Componentes del Box Model

- **Content** → el contenido (texto, imagen)
  - **Padding** → espacio interno
  - **Border** → borde de la caja
  - **Margin** → espacio externo
- 





# Padding y Margin

## Padding


Espacio **dentro** del elemento, entre el contenido y el borde.

```
div {  
  padding: 20px;  
}
```

## Margin

Espacio **fuera** del elemento, separándolo de otros.

```
div {  
  margin: 15px;  
}
```





# Bordes y dimensiones

- **Propiedad width y height**

Definen el tamaño del elemento.


```
div {  
  width: 300px;  
  height: 150px;  
}
```

- **Bordes**

```
div {  
  border: 2px solid black;  
}
```

- **Bordes redondeados**

```
div {  
  border-radius: 10px;  
}
```







# Cascada y prioridad en CSS



## ¿Qué es la cascada?

CSS decide qué estilo aplicar cuando **varias reglas afectan al mismo elemento**.

## Orden de prioridad (básico)

- CSS en línea (inline)
  - CSS interno
  - CSS externo
  -  El más cercano al elemento **gana**.
- 




# Cascada y prioridad en CSS




## Ejemplo simple


```
p{  
  color: blue;  
}  
<p style="color: red;">Texto</p>
```

👉 El texto será **rojo**.








# Buenas prácticas básicas en CSS



## Recomendaciones

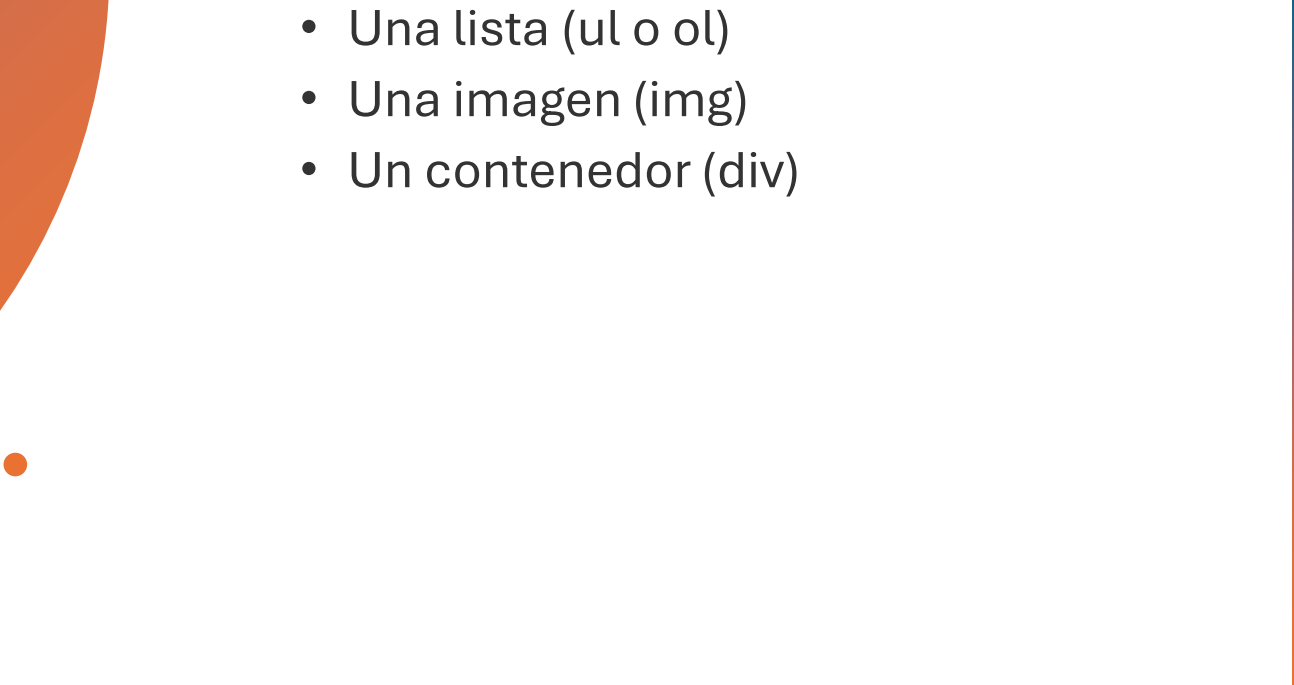
- Usar **CSS externo** siempre que sea posible
  - Evitar estilos inline
  - Usar **clases** para estilos reutilizables
  - Mantener el CSS ordenado y legible
- 




# Actividad práctica

## Actividad en clase / Tarea

Crear una página web utilizando **HTML + CSS externo** que contenga:

- Un título principal (h1)
  - Al menos dos párrafos (p)
  - Una lista (ul o ol)
  - Una imagen (img)
  - Un contenedor (div)
- 




# Actividad práctica



- **Requisitos de CSS**

- Aplicar:
- Colores de texto y fondo
- Tipografía básica
- Márgenes y padding
- Bordes y dimensiones

- **Restricciones**

-  No usar:
  - Flexbox
  - Diseño responsivo
  - Frameworks
  - Sass o preprocesadores
- 