



PROGRAMACIÓN

Explorador



Misión 1- Tema 2. Estilización y Diseño con CSS

John Edison Rodríguez Guisado

Ejecutor

Luis Gabriel Pacheco Tamayo

Mentor

G80 - UdeM

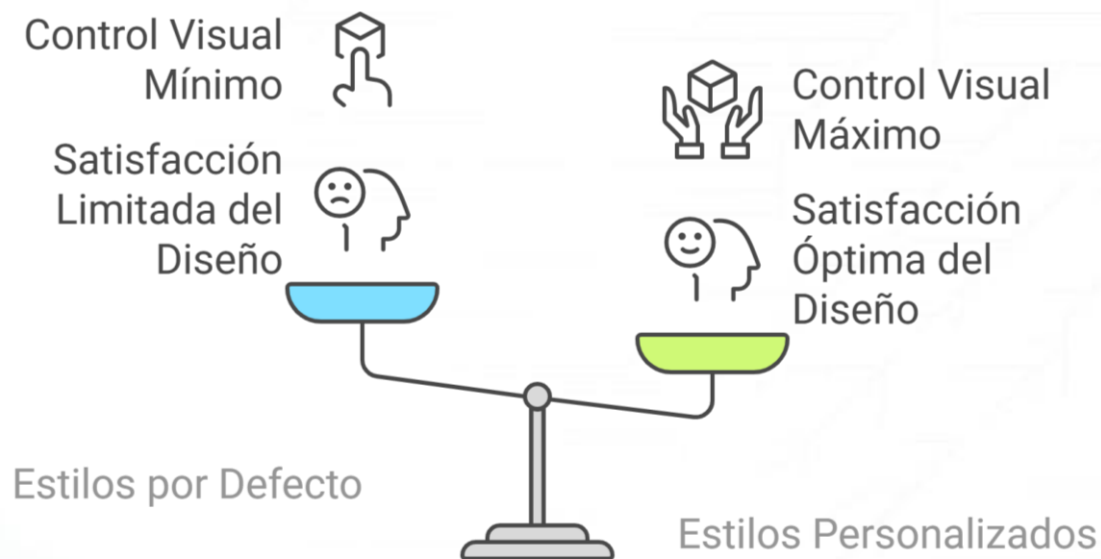
Introducción a CSS

CSS (Cascading Style Sheets), Hojas de Estilo en Cascada, es un lenguaje fundamental en el desarrollo web que se utiliza **en conjunto con HTML** para definir la **presentación visual** de los elementos en un documento.

CSS **permite** a los desarrolladores **aplicar estilos** como tamaño, color, fondo, bordes y más, mejorando así la estética y la usabilidad de las **páginas web**.



Estilo y estructura



A pesar de que cada navegador garantiza estilos por defecto para cada uno de los elementos **HTML**, estos estilos **no satisfacen** los requerimientos de cada diseñador.

Normalmente se encuentran muy distanciados de lo que queremos para nuestros sitios webs.

Diseñadores y desarrolladores a menudo **deben aplicar sus propios estilos** para obtener la **organización y el efecto visual** que realmente desean.



Estructura de Elementos en HTML: Block vs Inline

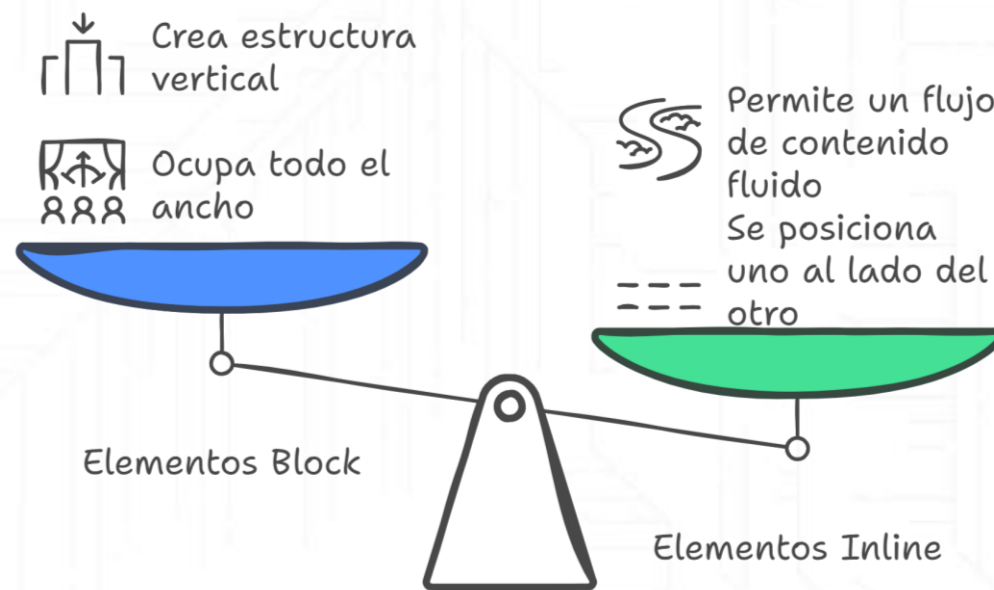
Con respecto a **la estructura**, básicamente cada navegador **ordena los elementos por defecto** de acuerdo a su tipo: **block (bloque)** o **inline (en línea)**. Esta clasificación está asociada con la forma en que los elementos son **mostrados en pantalla**.

Elementos block son posicionados uno sobre otro hacia abajo en la página.

Elementos inline son posicionados lado a lado, uno al lado del otro en la misma línea, sin ningún salto de línea a menos que ya no haya más espacio horizontal para ubicarlos.

Casi todos los elementos estructurales en nuestros documentos serán tratados por los navegadores como elementos block por defecto. Esto significa que cada **elemento HTML** que representa una parte de la **organización visual**.

Por ejemplo: <section>, <nav>, <header>, <footer>, <div>) será posicionado debajo del anterior.



Modelos de caja



Para aprender cómo podemos crear nuestra propia organización de los elementos en pantalla, debemos primero entender cómo los **navegadores procesan el código HTML**.

Los navegadores consideran cada **elemento HTML como una caja**.

Una **página web** es en realidad un **grupo de cajas ordenadas** siguiendo ciertas reglas.

Estas reglas son establecidas por estilos provistos por los navegadores o por los diseñadores **usando CSS**.

Propiedades de estilo CSS

Estilos en línea

Una de las técnicas más simples para incorporar estilos CSS a un documento HTML es la de asignar los estilos dentro de las etiquetas por medio del atributo style.

```
<!DOCTYPE html>
<html lang="es">

<head>
  <title> Este es el titulo del documento</title>
</head>

<body>
  <p style="font-size: 20px;">MI ESTILO DE
  LÍNEA</p>
</body>

</html>
```



Estilos embebidos

Una mejor alternativa es insertar los estilos en la cabecera del documento y luego usar referencias para afectar los elementos HTML correspondientes.

```
<!DOCTYPE html>
<html lang="es">

<head>
  <title> Este es el titulo del documento </title>
  <style>
    p {font-size: 20px}
  </style>
</head>

<body>
  <p> ESTE ES MI ESTILO DE LÍNEA </p>
</body>

</html>
```



Archivos externos

Declarar los **estilos en la cabecera** del documento ahorra espacio y vuelve al código más consistente y actualizable, pero **nos requiere hacer una copia de cada grupo de estilos en todos los documentos** de nuestro sitio web.

La solución es **mover todos los estilos a un archivo externo** y luego utilizar el elemento **<link>** para insertar este archivo dentro de cada documento que los necesite.

Este método **nos permite cambiar los estilos** por completo simplemente **incluyendo un archivo** diferente. También nos permite modificar o adaptar nuestros documentos a cada circunstancia o dispositivo

```
<!DOCTYPE html>
<html lang="es">

<head>
  <title> Este es el titulo del documento
</title>
  <link rel="stylesheet" href="misestilos.css">
</head>

<body>
  <p>ESTE ES MI ESTILO DE LÍNEA</p>
</body>

</html>
```



Uso de selectores CSS: clases, IDs y pseudo- clases

Referencia por Clase

Selecciona elementos por su nombre de clase



Referencia por Palabra Clave

Selecciona elementos por su nombre de etiqueta HTML



Referencia por ID

Selecciona elementos por su identificador único



Existen varios métodos para seleccionar cuáles elementos HTML serán afectados por las reglas CSS:

- Referencia por la **palabra clave** del elemento
- Referencia por el atributo **id**
- Referencia por el atributo **class**



Referenciando con palabra clave

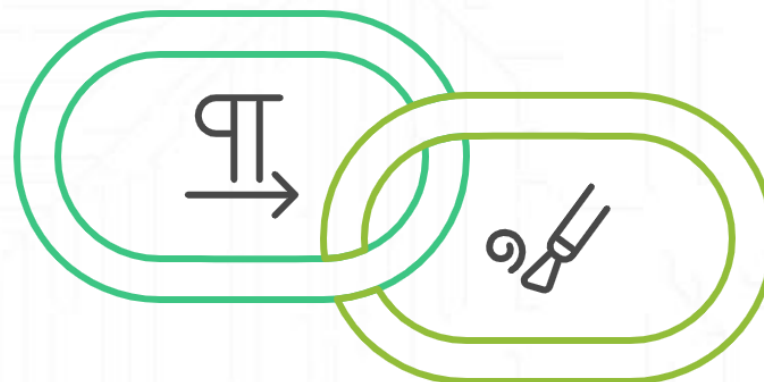
Al declarar las reglas CSS utilizando la palabra clave del elemento afectamos cada elemento de la misma clase en el documento.

```
p { font-size:20px }
```

```
span {font-size:20px}
```

Elemento p

Define el estilo para todos los elementos ``<p>`` en el documento.



Elemento span

Define el estilo para todos los elementos ```` en el documento.



Referenciando con el atributo id

```
<!DOCTYPE html>
<html lang="es">

<head>
  <title> Este es el titulo del documento
</title>
  <link rel="stylesheet" href="misestilos.css">
</head>

<body>
  <p id="texto1">ESTE ES MI ESTILO DE LÍNEA</p>
</body>

</html>
```

El atributo id es como un *nombre que identifica al elemento*. Esto significa que el valor de este atributo no puede ser duplicado. Este nombre debe ser único en todo el documento.

Para referenciar un elemento en particular usando el atributo id desde nuestro archivo CSS la regla debe ser *declarada* con el *símbolo #* al frente del valor que usamos para identificar el elemento

```
#texto1 { font-size: 20px }
```



Referenciando con el atributo class

La mayoría del tiempo, en lugar de utilizar el atributo id para propósitos de estilos es mejor utilizar class.

Este atributo es más flexible y puede ser asignado a cada elemento HTML en el documento que comparte un diseño similar.

```
.texto1 { font-size: 20px}
```

```
<!DOCTYPE html>
<html lang="es">

<head>
  <title> Este es el titulo del documento </title>
  <link rel="stylesheet" href="misestilos.css">
</head>

<body>
  <p class="texto1">ESTE ES MI ESTILO DE LÍNEA</p>
  <p class="texto1">ESTE ES MI ESTILO DE LÍNEA</p>
  <p>ESTE ES MI ESTILO DE LÍNEA</p>
</body>

</html>
```



Referenciando con cualquier atributo

```
p[name] { font-size: 20px }
```

Cambia solo elementos <p> que tienen un atributo llamado ***name***

```
p[name="mitexto"] { font-size: 20px }
```

Referenciando elementos <p> que tienen un atributo ***name*** con el valor ***mitexto***.

Aunque los métodos de referencia anteriores cubren un variado espectro de situaciones, a veces no son suficientes para encontrar el elemento exacto.

La última versión de CSS incorpora el **Selector de Atributo**.

Ahora podemos referenciar un elemento no solo por los atributos id y class sino también a través de cualquier otro atributo.



Referenciado atributo y RegEx

CSS3 permite combinar “=” con otros para hacer una selección más específica:

La regla con el **selector** ^= será asignada a todo elemento <p> que contiene un atributo name con un valor comenzado en “mi” (por ejemplo, “mitexto”, “micasa”).

La regla con el **selector** \$= será asignada a todo elemento <p> que contiene un atributo name con un valor finalizado en “mi” (por ejemplo “textomi”, “casami”).

La regla con el **selector** *= será asignada a todo elemento <p> que contiene un atributo name con un valor que incluye el texto “mi” (en este caso, el texto podría también encontrarse en el medio, como en “textomicasa”).

```
p[name^="mi"] { font-size: 20px }
```

```
p[name$="mi"] { font-size: 20px }
```

```
p[name*="mi"] { font-size: 20px }
```



Referenciando con pseudo clases

```
<!DOCTYPE html>
<html lang="es">

<head>
  <title> Este es el titulo del documento </title>
  <link rel="stylesheet" href="misestilos.css">
</head>

<body>
  <div id="wrapper">
    <p class="mitexto1">ESTE ES MI ESTILO DE LÍNEA 1</p>
    <p class="mitexto2">ESTE ES MI ESTILO DE LÍNEA 2</p>
    <p class="mitexto3">ESTE ES MI ESTILO DE LÍNEA 3</p>
    <p class="mitexto4">ESTE ES MI ESTILO DE LÍNEA 4</p>
  </div>
</body>

</html>
```

CSS3 también incorpora nuevas ***pseudo clases*** que hacen la selección aún más específica.

Usando pseudo clases podemos aprovechar esta **organización** y **referenciar un elemento específico** sin importar cuánto conocemos sobre sus atributos y el valor de los mismos.

```
p:nth-child(2) {
  background: #999999;
}
```



Pseudo Clases

Una pseudoclase CSS es una palabra clave que se añade a los selectores y que especifica un estado especial del elemento seleccionado.

:active => representa un elemento (como un botón) que el usuario está activando

:checked => representa cualquier radio, checkbox u option que está marcado o conmutado a un estado on

:default => representa cualquier elemento de formulario que sea el predeterminado entre un grupo de elementos relacionados

:focus => representa un elemento que ha recibido el foco

:hover => ocurre cuando el usuario interactúa con un elemento con un dispositivo señalador, pero no necesariamente lo activa.

:visited => representa enlaces que el usuario ya ha visitado

selector:pseudoclase { propiedad: valor; }



Ejercicio



Realiza el siguiente menú, empleando listas HTML y eventos del mouse para crear dinamismo con el cursor.

Solución 1: Referenciando con palabra clave

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8"> <!-- Codificación de caracteres -->
  <title>Menú con etiquetas HTML</title> <!-- Título de la página -->
  <style>
    /* Aplica estilo al contenedor de la lista */
    ul {
      background-color: #d8aee3; /* Color de fondo lila */
      padding: 15px;             /* Espaciado interno */
      margin: 0;                 /* Quita margen por defecto */
      list-style: none;          /* Elimina viñetas */
      display: flex;             /* Alinea los <li> en fila */
      gap: 20px;                /* Espacio entre ítems */
    }
  </style>
</head>
```





```
/* Estilos aplicados a todos los <li> del documento */  
li {  
    color: white;           /* Color del texto */  
    padding: 10px 20px;    /* Relleno interno del botón */  
    font-family: Georgia, serif; /* Fuente serif elegante */  
    font-size: 20px;       /* Tamaño de letra */  
    cursor: default;       /* Cursor normal inicialmente */  
    transition: background-color 0.3s, cursor 0.3s; /* Animación suave */  
}
```

```
/* Efecto al pasar el mouse por cualquier <li> */  
li:hover {  
    background-color: blue; /* Fondo azul al pasar el mouse */  
    cursor: pointer;       /* Cambia el cursor a mano */  
    font-weight: bold;     /* Negrita para resaltar */  
}
```

```
</style>
```

```
</head>
```



`<body>`

`<!-- Lista del menú con elementos sin clases ni IDs -->`

``

`Inicio`

`¿Quiénes Somos?`

`Contacto`

``

`<!-- Primer botón -->`

`<!-- Segundo botón -->`

`<!-- Tercer botón -->`

`</body>`

`</html>`

Ventajas:

- **Simplicidad:** No se necesitan clases, si todos los ítems son iguales.
- **Menos códigos:** CSS se aplica directamente a los elementos.
- **Claridad semántica:** Usar `` y `` es correcto para listas de navegación.



Solución 2: Referenciando con el atributo class

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8"> <!-- Establece codificación de caracteres -->
  <title>Menú Interactivo</title> <!-- Título de la pestaña del navegador -->
  <style>
    /* Contenedor general del menú con fondo lila */
    .menu-container {
      background-color: #d8aee3; /* Color de fondo */
      padding: 15px; /* Espaciado interno */
    }
  </style>
</head>
```





```
/* Lista del menú alineada horizontalmente */
.menu-list {
  list-style: none; /* Elimina viñetas */
  margin: 0;        /* Elimina margen por defecto */
  padding: 0;        /* Elimina relleno por defecto */
  display: flex;     /* Organiza elementos en línea */
  gap: 20px;         /* Espacio entre los elementos */
}

/* Estilo de cada elemento de la lista del menú */
.menu-item {
  color: white;      /* Color del texto */
  padding: 10px 20px; /* Espaciado interno del botón */
  font-family: Georgia, serif; /* Fuente serif */
  font-size: 20px;   /* Tamaño de texto */
  cursor: default;    /* Cursor normal por defecto */
  transition: background-color 0.3s, cursor 0.3s; /* Transiciones suaves */
}
```





```
/* Efecto al pasar el cursor sobre un ítem del menú */  
.menu-item:hover {  
  background-color: blue; /* Fondo azul al pasar el mouse */  
  cursor: pointer;        /* Cambia el cursor a mano */  
  font-weight: bold;       /* Texto en negrita */  
}  
</style>  
</head>
```



<body>

<!-- Contenedor del menú -->

<nav class="menu-container">

<!-- Lista no ordenada del menú -->

<ul class="menu-list">

<!-- Elementos del menú con clase para aplicar estilos -->

<li class="menu-item">Inicio

<!-- Primer botón -->

<li class="menu-item">¿Quienes Somos?

<!-- Segundo botón -->

<li class="menu-item">Contacto

<!-- Tercer botón -->

</nav>

</body>

</html>



Solución 3: Referenciando con el atributo id

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8"> <!-- Define la codificación del documento -->
  <title>Menú con ID</title> <!-- Título que aparece en la pestaña del navegador -->
  <style>
    /* Estilo del contenedor del menú */
    #menu-container {
      background-color: #d8aee3; /* Fondo color lila */
      padding: 15px; /* Espaciado interno */
    }
  </style>
</head>
```



```

/* Estilo para la lista del menú */
#menu-list {
  list-style: none; /* Elimina los puntos de la lista */
  margin: 0;        /* Quita el margen por defecto */
  padding: 0;        /* Quita el relleno por defecto */
  display: flex;     /* Muestra los elementos en una sola fila */
  gap: 20px;         /* Espacio entre los botones */
}

/* Estilo común para cada ítem del menú */
.menu-item {
  color: white;       /* Color de texto blanco */
  padding: 10px 20px; /* Espaciado interno del botón */
  font-family: Georgia, serif; /* Fuente serif */
  font-size: 20px;    /* Tamaño de fuente */
  cursor: default;    /* Cursor normal por defecto */
  transition: background-color 0.3s, cursor 0.3s; /* Transición suave al mouse */
}

```





```
/* Cambios visuales al pasar el cursor por un ítem */  
.menu-item:hover {  
  background-color: blue; /* Cambia el fondo a azul */  
  cursor: pointer;        /* Cambia el cursor a mano */  
  font-weight: bold;       /* Resalta el texto */  
}  
</style>  
</head>
```





<body>

<!-- Contenedor principal del menú -->

<nav id="menu-container">

<!-- Lista no ordenada que contiene los ítems del menú -->

<ul id="menu-list">

<!-- Cada ítem usa la clase .menu-item pero se gestiona desde un contenedor con ID -->

<li class="menu-item" id="inicio">Inicio

<!-- Primer botón -->

<li class="menu-item" id="quienes">¿Quienes Somos?

<!-- Segundo botón -->

<li class="menu-item" id="contacto">Contacto

<!-- Tercer botón -->

</nav>

</body>

</html>

Que cambio con respecto al anterior:

- Se usan atributos id para identificar los bloques principales.
- Se mantienen el uso de de clases para dar estilos a los ítem, ya que por norma los atributos id deben de ser únicos en todo el documento.
- Se presenta combinación de ambos referenciados.



GRACIAS