

# Módulo 4: CSS

## 1. Introducción

En esta sección se presenta el lenguaje CSS (Cascading Style Sheets), empleado para definir la apariencia visual de los elementos HTML. Se explica cómo CSS complementa al HTML para lograr diseños atractivos y funcionales.

---

## 2. Sintaxis básica

CSS utiliza una sintaxis sencilla basada en **selectores** y declaraciones. Un selector identifica el elemento HTML que se desea estilizar, mientras que las declaraciones, encerradas entre llaves, especifican las propiedades y sus valores.

---

## 3. Selectores

Los selectores permiten apuntar a elementos específicos del HTML para aplicarles estilos. Se utilizan selectores de tipo (por ejemplo, encabezados o párrafos), de clase (identificados con un punto) e ID (identificados con el símbolo de numeral). Además, existen selectores de atributos y combinados que facilitan la selección de elementos según su relación o características particulares.

---

## 4. Comentarios

Los comentarios en CSS se escriben entre `/*` y `*/`. Son útiles para dejar anotaciones o explicaciones en el código sin que afecten su funcionamiento.

---

## 5. Colores

CSS permite definir colores mediante diferentes formatos: nombres, valores hexadecimales, RGB, entre otros. La elección de colores es fundamental para el diseño visual y la identidad de la página.

---

## 6. Background (Fondo)

La propiedad de fondo se usa para establecer el color, imagen o gradiente que se aplicará como fondo de un elemento. Se pueden ajustar parámetros como la posición, el tamaño y la repetición del fondo.

---

## **7. Bordes**

Con la propiedad de bordes se definen los límites de un elemento. Se especifica el grosor, el estilo (sólido, punteado, etc.) y el color, lo cual es útil para resaltar o delimitar secciones.

---

## **8. Unidades de medida**

Las unidades en CSS permiten definir tamaños y espacios. Entre las más comunes se encuentran los píxeles (px), las unidades relativas como em y rem, y el porcentaje (%). Cada unidad se utiliza en función de la necesidad de un diseño responsivo o fijo.

---

## **9. Margin (Márgenes)**

La propiedad margin establece el espacio exterior de un elemento, definiendo la separación entre el elemento y sus vecinos en la página.

---

## **10. Padding**

El padding define el espacio interior de un elemento, es decir, el espacio entre el contenido y el borde del elemento. Permite que el contenido respire y mejore la legibilidad.

---

## **11. Alto y Ancho (Height & Width)**

Mediante las propiedades height y width se determinan las dimensiones de un elemento. Estos valores pueden ser fijos o relativos, adaptándose a distintos diseños.

---

## **12. Texto**

CSS ofrece numerosas propiedades para el estilado del texto, como la familia tipográfica, el tamaño, el interlineado y la alineación. Estos atributos son esenciales para asegurar la legibilidad y coherencia visual del contenido.

---

### **13. Fuentes**

Con la propiedad `font-family` se seleccionan las fuentes tipográficas, mientras que propiedades adicionales como `font-style` y `font-weight` permiten definir estilos y grosores que aporten personalidad al diseño.

---

### **14. Íconos**

Los íconos complementan el diseño visual y la funcionalidad. Se pueden integrar mediante imágenes o utilizando bibliotecas especializadas (como Font Awesome), que ofrecen íconos escalables y personalizables.

---

### **15. Listas**

Las listas pueden ser ordenadas o desordenadas y se estilizan mediante la propiedad `list-style`, permitiendo ajustar la viñeta, numeración o apariencia de los elementos listados.

---

### **16. Tablas**

Las tablas se estilizan a través de propiedades que controlan la apariencia de celdas, filas y columnas. Propiedades como `border-collapse`, `text-align` y `padding` son fundamentales para un diseño claro y ordenado.

---

### **17. Display & Visibility**

Las propiedades `display` y `visibility` determinan la forma en que los elementos se presentan en la página. Mientras que `display` puede cambiar el comportamiento (por ejemplo, `block`, `inline` o `flex`), `visibility` controla si el elemento es visible sin afectar el flujo del diseño.

---

### **18. Position**

La propiedad `position` gestiona el posicionamiento de los elementos, pudiéndose establecer en valores como `static`, `relative`, `absolute` o `fixed`. Esto es clave para diseños más complejos que requieren una colocación precisa.

---

## **19. Z-Index**

Con z-index se controla el orden de apilamiento de los elementos posicionados, permitiendo que ciertos elementos se superpongan a otros según se requiera en el diseño.

---

## **20. Overflow**

La propiedad overflow define qué ocurre cuando el contenido excede el espacio asignado a un elemento. Puede configurarse para ocultar el contenido excedente, mostrar barras de desplazamiento o ajustarlo automáticamente.

---

## **21. Float**

Float permite que un elemento se desplace hacia un lado (izquierda o derecha) dentro de su contenedor, haciendo que el texto u otros elementos fluyan a su alrededor. Es una técnica común en el diseño de layouts.

---

## **22. Selectores combinados**

Estos selectores permiten aplicar estilos basados en la relación entre elementos. Por ejemplo, se puede seleccionar un elemento que es hijo de otro o que se encuentra adyacente a otro, facilitando estilos más precisos y contextualizados.

---

## **23. Pseudo Clases**

Las pseudoclasas aplican estilos a elementos en estados específicos, como cuando un usuario pasa el ratón sobre un enlace (:hover) o cuando un campo tiene el foco (:focus). Son fundamentales para la interacción y la experiencia de usuario.

---

## **24. Pseudo Elementos**

Permiten estilizar partes específicas de un elemento, como el contenido antes o después de un elemento, usando pseudo elementos como ::before y ::after. Esto resulta útil para agregar contenido decorativo sin alterar el HTML.

---

## 25. Opacidad

La propiedad `opacity` gestiona el nivel de transparencia de un elemento, permitiendo valores entre 0 (completamente transparente) y 1 (totalmente opaco).

---

## 26. Navegación (NavBar)

El diseño de una barra de navegación se logra combinando etiquetas HTML (como `<nav>`) y propiedades CSS. Se estudian métodos para organizar los elementos de forma horizontal o vertical, optimizando la experiencia de usuario en diferentes dispositivos.

---

## 27. Selector de Atributos

Este selector permite aplicar estilos a elementos que contengan determinados atributos o valores, facilitando una selección más específica y dinámica según las características del HTML.

---

## 28. `!important`

La declaración `!important` se utiliza para forzar la aplicación de una regla CSS, otorgándole la máxima prioridad en el cascada. Se recomienda su uso moderado para evitar conflictos en el código.

---

## 29. Animaciones

Las animaciones en CSS permiten cambiar gradualmente el estado de un elemento a lo largo del tiempo. Se definen mediante la regla `@keyframes`, que detalla las transformaciones que ocurrirán durante la animación.

---

## 30. Transiciones

Las transiciones facilitan el cambio suave entre diferentes estados de un elemento, creando efectos visuales agradables al modificar propiedades CSS en respuesta a acciones del usuario.

---

### **31. MediaQueries (Responsive)**

Las Media Queries permiten ajustar los estilos CSS según las características del dispositivo, como el tamaño de la pantalla. Esto es esencial para crear diseños responsivos que funcionan en móviles, tabletas y escritorios.

---

### **32. FlexBox**

FlexBox es un modelo de diseño que organiza los elementos de forma flexible dentro de un contenedor. Facilita la alineación, distribución y orden de los elementos en layouts complejos.

---

### **33. Grid**

El sistema Grid de CSS permite crear diseños estructurados en filas y columnas, ofreciendo un control preciso sobre la distribución y posicionamiento de los elementos en la página.

---

### **34. Buenas prácticas y accesibilidad**

Es fundamental seguir buenas prácticas en la escritura de CSS, como usar convenciones de nomenclatura (por ejemplo, BEM) y mantener el código modular y limpio. Además, se debe prestar atención a la accesibilidad, utilizando contrastes adecuados, tipografías legibles y un diseño que facilite la navegación para usuarios con diferentes capacidades.

---

## **Proyecto 1 - Estilado CSS**

En esta fase, se profundiza en el estilado del proyecto, aplicando propiedades y técnicas de CSS para mejorar la apariencia visual y la experiencia del usuario.

---

## **Proyecto 1 - Github - Netlify**

Finalmente, se guía a los estudiantes en el despliegue de su proyecto en Netlify, haciendo que la página web sea accesible en línea.