HOJAS DE ESTILO EN LA CONSTRUCCIÓN DE PÁGINAS WEB

Introducción a las Hojas de Estilo en la Construcción de Páginas Web

INTRODUCCIÓN A LAS HOJAS DE ESTILO

- Las hojas de estilo en cascada (CSS) son una herramienta fundamental para definir la presentación y el diseño de las páginas web.
- Permiten separar el contenido HTML de su apariencia visual, facilitando una mayor flexibilidad y mantenimiento.

FUNCIONES Y VENTAJAS DE LAS HOJAS DE ESTILO

- **Define:**
  - Colores, tipografías, espacios y distribuciones.
  - Animaciones y transiciones.
- **Ventajas:**
  - Reutilización de código y mantenimiento sencillo.
  - Separación de estructura y presentación.
  - Mejora de la accesibilidad y optimización del rendimiento.

TIPOS DE HOJAS DE ESTILO

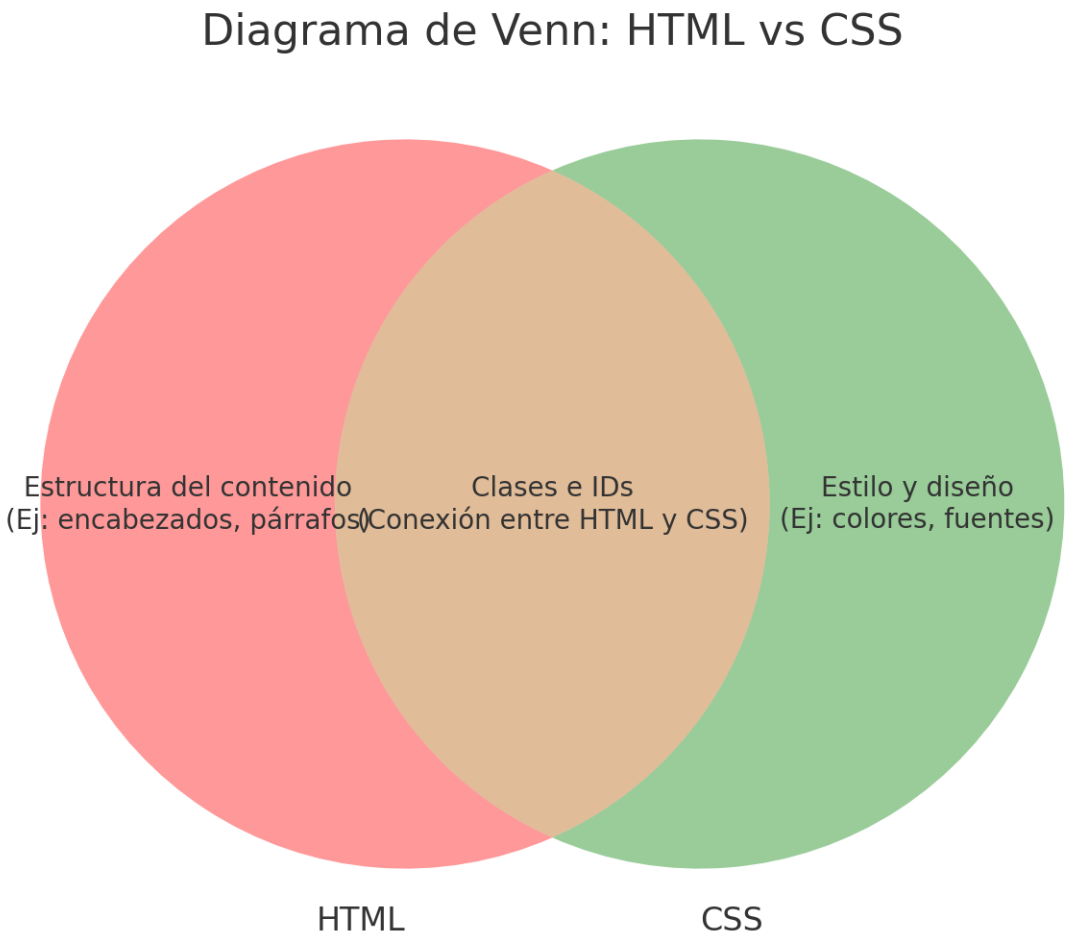
- **Estilos en línea:**
  - Se aplican directamente en las etiquetas HTML (<p style="color:red;">).
  - **Desventajas:** Dificulta el mantenimiento y la reutilización.
- **Estilos incrustados:**
  - Dentro de la etiqueta <style> en el <head>.
  - **Útil para una página específica.**
- **Estilos enlazados:**
  - Archivo CSS externo (estilo.css) enlazado con <link>.
  - **Ideal para aplicar estilos globales a múltiples páginas.**
- **Estilos importados:**
  - Usan la regla @import url('estilo.css'); en otro CSS.
  - Permite dividir CSS en módulos.

TIPOS DE HOJAS DE ESTILO

Tipo	Descripción	Ejemplo
En línea	Dentro de una etiqueta HTML	<p style="color: blue;">
Incrustado	Dentro de la sección <head> del HTML	<style> p { color: blue; }</style>
Enlazado	En un archivo CSS externo	<link rel="stylesheet" href="styles.css">
Importado	Importando otro archivo CSS	@import url('styles.css');

SEPARACIÓN DE CONTENIDO Y PRESENTACIÓN

- **HTML:** Se encarga de la estructura y contenido de la página.
- **CSS:** Se encarga del aspecto visual y la presentación.
- **Ventajas:**
  - Mantenimiento más fácil.
  - Mejora de la accesibilidad.
  - Carga más rápida y mejor posicionamiento en buscadores.



MANTENIMIENTO Y REUTILIZACIÓN DE CÓDIGO

- **Cambiar estilos de forma global** desde un archivo CSS.
- Al utilizar hojas de estilo enlazadas, es posible realizar cambios en una sola hoja de estilo y ver cómo se aplican de manera uniforme en todo el sitio.
- **Ahorro de tiempo y consistencia visual**.
- **Ejemplo:**

```
body {
    font-family: Arial, sans-serif;
    background-color: #f0f0f0;
}
```

SINTAXIS BÁSICA DE CSS

- La sintaxis de CSS consiste en selectores y reglas.

```
p {
    color: blue;
    font-size: 14px;
}
```

- **Selector:** Especifica el elemento HTML (p).
- **Propiedad:** Característica a modificar (color).
- **Valor:** Configuración aplicada (blue).

SELECTORES EN CSS

- **Selectores de tipo**: Seleccionan todos los elementos de un tipo específico (**h1**, **p**, **div**).
- **Selectores de clase**: Utilizan el carácter **.** seguido del nombre de la clase (**.boton**).
- **Selectores de ID**: Utilizan el carácter **#** seguido del identificador del elemento (**#encabezado**).
- **Selectores de atributo**: Seleccionan elementos en función de la presencia o valor de un atributo específico (**input[type="text"]**).
- **Selectores de descendiente:** Seleccionan elementos que son descendientes de otro elemento (div p).

- **Selectores de tipo:** p, h1.
- **Selectores de clase:** .boton {}.
- **Selectores de ID:** #encabezado {}.
- **Selectores de atributo:** input[type="text"] {}.
- **Selectores descendientes:** div p {}.

ESPECIFICIDAD EN CSS

- La especificidad determina qué reglas se aplican a un elemento cuando hay múltiples reglas que podrían aplicarse.
- **Ejemplo**: Los selectores de ID tienen una mayor especificidad que los selectores de clase.
- **Especificidad:** Determina qué regla se aplica cuando hay conflictos.

- **Prioridad:**
  1. Estilos en línea.
  2. Selectores de ID.
  3. Selectores de clase.
  4. Selectores de tipo.
- **Modelo de cascada:** El orden importa.

CASCADA Y HERENCIA

- **Cascada**: Proceso mediante el cual las reglas se aplican en un orden específico dependiendo de la especificidad y el origen de las reglas.
- **Herencia**: Permite que ciertos estilos aplicados a un elemento padre se transmitan automáticamente a sus elementos hijos.

BUENAS PRÁCTICAS EN EL USO DE CSS

- **Reutilización**: Utilizar clases en lugar de estilos en línea.
- **Organización**: Dividir el archivo CSS en secciones lógicas.
- **Modularización**: Utilizar archivos CSS separados para diferentes propósitos.
- **Uso de Variables**: Utilizar variables CSS (Custom Properties) para definir valores que se utilizan en múltiples lugares.
- **Consistencia**: Mantener una consistencia en la nomenclatura de clases y en el uso de unidades.

HERRAMIENTAS PARA TRABAJAR CON CSS

- **Preprocesadores**: Sass, Less.
- **Herramientas de Depuración**: Chrome DevTools.
- **Beneficios**: Identificar y corregir problemas, optimizar la presentación de la página web.

SINTAXIS Y ESTRUCTURA DE LOS ESTILOS EN CSS

ESTRUCTURA BÁSICA DE UNA REGLA CSS

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

- **Selector:** p (Selecciona todos los párrafos).
- **Propiedades:** color, font-size.
- **Valores:** red, 16px.
- Las reglas se agrupan en bloques {}.

TIPOS DE SELECTORES 1/2

Selector Universal (*): Selecciona todos los elementos del documento.

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

Selectores de Tipo: Seleccionan todos los elementos de un tipo específico.

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

Selectores de Clase (.): Seleccionan los elementos que tienen una clase específica.

```
.destacado {
    color: blue;
    font-weight: bold;
```

TIPOS DE SELECTORES 2/2

Selectores de ID (#): Seleccionan un único elemento que tiene un ID específico.

```
#principal {
    background-color: lightgrey;
    padding: 20px;
}
```

Selectores de Atributo: Seleccionan elementos que tienen un atributo específico o un valor específico de atributo.

```
input[type="text"] {
    border: 1px solid black;
}
```

COMBINADORES EN CSS (1/2)

- **Selector Descendiente ( ):** Selecciona los elementos que son descendientes de otro elemento.

  ```
  div p {
      color: green;
  }
  ```

  Afecta a **todos los párrafos dentro de un div**.

- **Selector de Hijo Directo (>):** Selecciona únicamente los hijos directos de un elemento.

  ```
  ul > li {
      list-style-type: square;
  }
  ```

  Afecta solo a **los hijos directos**.

COMBINADORES EN CSS (2/2)

- **Elemento Adyacente (+):** Selecciona el primer elemento que es adyacente a otro.

```
h2 + p {
    margin-top: 0;
}
```

Afecta al **primer párrafo después de un h2**.

- **Selector de Hermano General (~):** Selecciona todos los elementos que son hermanos de un elemento específico.

```
h2 ~ p {
    color: darkred;
}
```

Afecta a **todos los párrafos que sean hermanos de un h2**.

PSEUDO-CLASES

Definen un estado especial de un elemento.

- **:hover**

  a:hover {

  text-decoration: underline;

  }

  Aplica cuando el **ratón pasa por encima de un enlace**.

- **:focus**

  input:focus {

  border-color: blue;

  }

  Aplica cuando un **campo está activo**.

Las pseudo-clases permiten definir estilos dinámicos basados en interacciones del usuario

PSEUDO-ELEMENTOS

Permiten aplicar estilo a una parte específica de un elemento.

- **::before**

  p::before {

  content: " 🔹 ";

  }

  Añade contenido **antes del párrafo**.

- **::after**

  p::after {

  content: " 🔹 ";

  }

  Añade contenido **después del párrafo**.

Los pseudo-elementos permiten añadir elementos decorativos sin modificar el HTML.

COMENTARIOS EN CSS

```
/* Este es un comentario en CSS */
body {
    background-color: white;
}
```

- Documentan el código.
- Facilitan el mantenimiento.

BUENAS PRÁCTICAS PARA ESCRIBIR CSS

- **Mantener el Código Limpio y Ordenado**: Dividir el CSS en secciones con comentarios, y evitar reglas repetitivas.
- **Uso Moderado de Selectores Universales y Específicos**: Utilizar selectores eficientes que no impacten negativamente en el rendimiento del sitio.
- **Evitar la Sobreespecificación**: Intentar que los selectores no sean innecesariamente complejos.
- **Agrupación de Reglas**: Agrupar elementos que compartan reglas comunes para evitar duplicación de código.

```
h1, h2, h3 {
    font-family: Arial, sans-serif;
    color: navy;
}
```

ATRIBUTOS DE ESTILO PARA FUENTES, COLORES Y FONDO

ESTILOS DE FUENTES – INTRODUCCIÓN

- Las fuentes son un componente importante en el diseño de una página web.
- CSS permite controlar varios aspectos de las fuentes.
  - **Familia de fuentes (font-family).**
  - **Tamaño de fuente (font-size).**
  - **Grosor del texto (font-weight).**
  - **Estilo del texto (font-style).**
  - **Altura de línea (line-height).**

PROPIEDAD FONT-FAMILY

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

- **Define la fuente del texto.**
- Se recomienda usar **fuentes seguras** (Arial, Verdana) y **familias de respaldo**.
- **Ejemplo:** Arial, si no está disponible, se usa Helvetica o una fuente genérica sans-serif.

PROPIEDAD FONT-SIZE

```
h1 {
    font-size: 2em;
}
```

- **Define el tamaño del texto.**
- **Unidades absolutas:** px, pt.
- **Unidades relativas:** em, rem, %, vw.
- **Recomendación:** Utilizar **unidades relativas (em, %)** para mejorar la accesibilidad.

PROPIEDADES FONT-WEIGHT Y FONT-STYLE

- **font-weight:** Controla el grosor del texto.

```
h2 {
    font-weight: bold;
}
```

- **font-style:** Define si el texto es cursiva o normal.

```
em {
    font-style: italic;
}
```

- **Valores:** normal, bold, bolder, lighter, o números de 100 a 900.

PROPIEDAD LINE-HEIGHT

- Controla el espacio entre líneas de texto.

```
p {
    line-height: 1.5;
}
```

ESTILOS DE COLORES – INTRODUCCIÓN

- Los colores se definen usando:
  - **Nombres de color:** red, blue.
  - **Hexadecimal:** #ff0000.
  - **RGB y RGBA:** rgb(255, 0, 0), rgba(255, 0, 0, 0.5).
- **Ejemplo:**

```
p {
    color: #333;
}
```

PROPIEDAD BACKGROUND-COLOR

- Define el color de fondo de un elemento.
- **Ejemplo**:

```
body {
    background-color: #f0f0f0;
}
```

PROPIEDAD OPACITY

- Controla la opacidad de un elemento.
- **Ejemplo**:

```
div {
    opacity: 0.8;
}
```

- **opacity:** Ajusta la transparencia (de 0 a 1).

FONDOS EN CSS

- El fondo de un elemento puede personalizarse utilizando varias propiedades de CSS.

PROPIEDAD BACKGROUND-IMAGE

- Permite añadir una imagen de fondo a un elemento.
- **Ejemplo**:

```
body {
    background-image: url('imagen.jpg');
}
```

PROPIEDAD BACKGROUND-REPEAT

- Controla si la imagen de fondo se repite.
- **Ejemplo**:

```
div {
    background-repeat: no-repeat;
}
```

- Se pueden usar (repeat, no-repeat, repeat-x, repeat-y).

PROPIEDAD BACKGROUND-POSITION

- Define la posición de la imagen de fondo.
- **Ejemplo**:

```
div {
    background-position: center;
}
```

- Los valores que se pueden usar (top, bottom, center, coordenadas específicas).

PROPIEDAD BACKGROUND-SIZE

- Controla el tamaño de la imagen de fondo.
- **Ejemplo**:

```
div {
    background-size: cover;
}
```

- Los valores que se pueden usar (cover, contain, dimensiones específicas).

BUENAS PRÁCTICAS CON FUENTES, COLORES Y FONDOS

- **Legibilidad:** Evitar combinaciones con bajo contraste.
- **Consistencia:** Mantener una paleta de colores coherente.
- **Sistema de Fuentes Seguro:** Definir alternativas de fuente (font-family).
- **Evitar el Exceso:** Máximo **dos o tres tipografías** por sitio.
- Es recomendable la creación de una **guía de estilo** con tipografías y colores definidos.

ATRIBUTOS DE ESTILO PARA BLOQUES Y TEXTOS

INTRODUCCIÓN A LOS ESTILOS DE BLOQUES

- Los bloques son **contenedores de contenido** en HTML.
- CSS permite controlar:
  - **Espacio exterior (margin).**
  - **Espacio interior (padding).**
  - **Bordes (border).**
  - **Dimensiones (width, height).**
- **Visualización (display).**

PROPIEDAD MARGIN

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

- **Define el espacio exterior del elemento.**
- Puede definirse:
  - De forma **individual**: margin-top, margin-right, margin-bottom, margin-left.
  - De forma **abreviada**: margin: 10px 20px 30px 40px; (arriba, derecha, abajo, izquierda).

PROPIEDAD PADDING

```
section {
    padding: 15px;
}
```

- **Define el espacio interior** entre el borde y el contenido del elemento.
- Se puede usar de manera:
  - **Individual**: padding-top, padding-bottom.
  - **Abreviada**: padding: 15px 30px; (vertical, horizontal).
- Aumenta el **área de interacción** del elemento.

PROPIEDAD BORDER

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

- **Crea un borde alrededor del elemento.**
- Sintaxis: border: [grosor] [estilo] [color];
- Estilos comunes:
  - solid (sólido), dotted (punteado), dashed (guiones).

DIMENSIONES CON WIDTH Y HEIGHT

```
.caja {
    width: 300px;
    height: 200px;
}
```

- **Define el ancho y alto de los elementos.**
- Unidades:
  - **Absolutas:** px, cm.
  - **Relativas:** %, em, rem.

PROPIEDAD DISPLAY

```
span {
    display: block;
}
```

- **Controla la visualización de los elementos.**
- Valores comunes:
  - block (ocupa toda la línea), inline (en línea con otros elementos), flex (flexible), grid (en cuadrícula).

ALINEACIÓN DE TEXTO - TEXT-ALIGN

```
p {
    text-align: justify;
}
```

- **Define la alineación del texto dentro de un bloque.**
- Valores:
  - left (izquierda), right (derecha), center (centrado), justify (justificado).

PROPIEDAD TEXT-DECORATION

- Controla los efectos decorativos del texto.
- **Ejemplo**:

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

- Los valores que se pueden usar (underline, line-through, overline, none).
- Subrayado, tachado (underline, line-through).

PROPIEDAD TEXT-TRANSFORM

- Controla la transformación del texto.
- **Ejemplo**:

```
h2 {
    text-transform: uppercase;
}
```

- Los valores que se pueden usar (uppercase, lowercase, capitalize).

PROPIEDAD LETTER-SPACING Y WORD-SPACING

- Permiten controlar el espacio entre letras y palabras.
- **Ejemplo**:

```
p {
    letter-spacing: 1.5px;
    word-spacing: 3px;
}
```

- Estas propiedades pueden mejorar la legibilidad del texto.

PROPIEDAD LINE-HEIGHT

- Controla la altura de línea.
- **Ejemplo**:

```
p {
    line-height: 1.8;
}
```

- Esta propiedad mejora la legibilidad del texto.

PROPIEDAD WHITE-SPACE

- Controla cómo se gestiona el espacio en blanco dentro de un elemento.
- **Ejemplo**:

```
p {
    white-space: nowrap;
}
```

- Los valores que se pueden usar (normal, nowrap, pre, pre-wrap, pre-line).