

«Talento Tech»

Front-End JS

Clase 04



Clase N° 4: Introducción a CSS

Temario:

1. Bases del CSS

- Concepto y utilidad de CSS
- Separación de estilos y estructura HTML
- Reutilización de estilos

2. Cómo Incorporar CSS

- CSS en Línea
- CSS Interno
- CSS Externo

3. Selectores Básicos

- Selector Universal
- Selector de Etiqueta
- Selector de Clase
- Selector de ID

4. Especificidad, Herencia, Cascada y Orden de las Reglas en CSS

- Especificidad
- Herencia
- Cascada
- Orden de las Reglas

5. Atributo básicos de css

- Color
- Tamaño
- Unidades de medida en css



1. Bases del CSS

CSS (Cascading Style Sheets u “*hojas de estilo en cascada*” en español) es la herramienta que usamos para dar vida a nuestras páginas web, dándoles color, formas y un diseño atractivo. Sin CSS, nuestras páginas serían simples bloques de texto e imágenes. Su principal función es controlar cómo se presentan los elementos HTML en el navegador. Pero, ¿qué es esto de la “**cascada**”? Este término hace referencia a cómo se resuelven los conflictos cuando varias reglas se aplican a un mismo elemento.

Importancia del CSS:

- **Separación de “concerns”:** CSS nos permite separar el contenido (HTML) de su presentación, lo que hace que todo sea más organizado. Imaginate tener todos los estilos mezclados con el HTML, ¡un lío total!
- **Reutilización de estilos:** Lo mejor de CSS es que los estilos que definimos en un archivo pueden ser utilizados en varias páginas, logrando así un diseño consistente sin repetir código y dando un mismo criterio estético a nuestras páginas web.

2. ¿Cómo incorporar CSS?

Podemos agregar CSS a nuestras páginas de tres maneras diferentes. Algunas son mejores que otras según la situación, pero te cuento sobre cada una de ellas.

2.1 CSS en línea

Es el más básico y fácil de aplicar, pero no es realmente recomendable porque “ensucia” el HTML. Se usa con el atributo `style` dentro de una etiqueta.

Ejemplo:

```
<p style="color: red; font-size: 14px;">Este párrafo es rojo y tiene un tamaño de catorce píxeles</p>
```

2.2 CSS interno

Esta metodología la utilizaremos cuando queramos aplicar estilos sólo a una página específica. Colocamos el CSS dentro de la etiqueta `<style>` en el `<head>` del documento.

Ejemplo:

```
<head>

<style>

    h1 {

        color: blue;

        background-color: yellow;

    }

</style>

</head>
```

2.3 CSS externo

El mejor y más utilizado. Guardamos el CSS en un archivo aparte y lo vinculamos con el HTML usando la etiqueta `<link>`. ¡Ideal para proyectos más grandes!

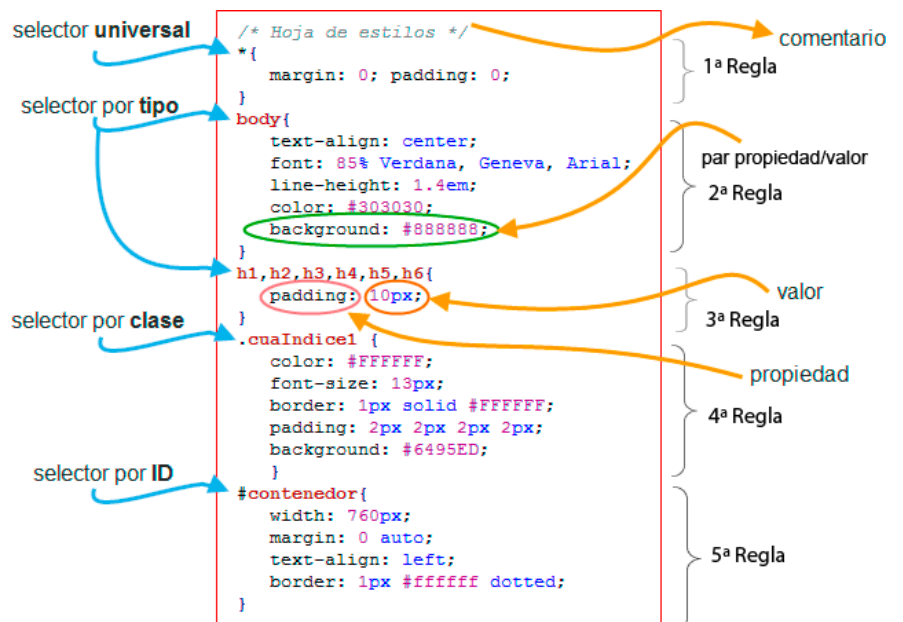
```
<head>

    <link rel="stylesheet" href="css/styles.css">

</head>
```

3. Selectores básicos

En CSS, los selectores nos permiten aplicar estilos a los elementos HTML que queramos, eligiendo específicamente a cuáles de ellos afectaremos con los estilos especificados. Hay varias formas de asignar estilos, a continuación explicaremos los más básicos.



3.1 Selector universal

Este selector (*) afecta a todos los elementos de la página. Es útil para aplicar reglas generales.

Ejemplo:

```
* {  
  
    margin: 0;  
  
    padding: 0;  
  
    box-sizing: border-box;  
  
}
```

3.2 Selector de etiqueta

Aplica estilos a todos los elementos de un tipo, por ejemplo, a todos los párrafos `<p>`.

Ejemplo:

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

3.3 Selector de clase

Si querés aplicar un estilo a varios elementos pero no a todos podés usar **clases**. Definimos una clase con un punto (.) y la aplicamos a los elementos que queramos.

Ejemplo:

```
.highlight {  
  
    background-color: yellow;  
  
}
```

3.4 Selector de ID

Para un estilo único, usamos un **ID**. Solo puede aplicarse a un elemento específico. Lo definimos con un numeral (**#**).

Ejemplo:

```
#main-title {  
  
    font-size: 24px;  
  
    color: green;  
  
}
```

4. Especificidad, herencia, cascada y orden de las reglas en CSS

A veces, varios estilos pueden estar aplicando al mismo elemento. En esos casos, el CSS utiliza un sistema de **especificidad y cascada** para decidir cuál es la indicación de mayor jerarquía que se aplicará sobre otras reglas que también estén operando.

4.1 Especificidad

Las reglas con mayor especificidad tienen más peso. Un estilo en línea (dentro de una etiqueta) tendrá mayor prioridad que una clase o un ID.

- **Estilo en línea:** `style="color:red;"`
- **ID:** `#id`
- **Clase:** `.clase`

4.2 Herencia

Algunas propiedades, como `color` y `font-family`, se heredan de los elementos “padres”. Por ejemplo, si definís el color de un `<div>`, todos los elementos dentro de ese div adoptarán ese color.

4.3 Cascada

Cuando varias reglas tienen la misma especificidad, se aplica la que esté más abajo en el archivo CSS.

4.4 Orden de las reglas

Si dos reglas afectan a un mismo elemento y tienen la misma especificidad, se aplicará la última.

Ejemplo:

```
p {  
  
    color: blue;  
  
}  
  
p {  
  
    color: red; /* Esta regla se aplicará porque está después */  
  
}
```

5. Atributos básicos en CSS

5.1 Color

El atributo color en CSS permite cambiar el color del texto de los elementos HTML. Hay varias formas de especificar colores en CSS, y es importante entender cómo funcionan. Las formas más comunes son:

Nombres de colores: CSS reconoce muchos nombres de colores, como red, blue, black, white, etc. Este método es útil cuando quieres usar un color estándar.

Ejemplo:

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

Valores hexadecimales: Los valores hexadecimales se componen de seis dígitos que representan los colores rojo, verde y azul (RGB). Por ejemplo, el rojo se escribe como #FF0000, donde FF es el valor máximo para el rojo.

Ejemplo:

```
h1 {  
  
    color: #3498db; /* Azul claro */  
  
}
```


RGB y RGBA: Con `rgb()` se pueden definir colores usando valores numéricos de 0 a 255 para los canales rojo, verde y azul. `rgba()` añade un valor de opacidad que va de 0 (totalmente transparente) a 1 (totalmente opaco).

Ejemplo:

```
h2 {  
  
    color: rgb(255, 99, 71); /* Tomate */  
  
}  
  
.semi-transparente {  
  
    color: rgba(255, 99, 71, 0.5); /* Tomate semitransparente */  
  
}
```

5.2 Tamaño

El tamaño de los elementos HTML, como textos o bloques, se define en CSS mediante diferentes propiedades como `font-size` (para el texto) o `width` y `height` (para el ancho y la altura de los contenedores).

`font-size`: Controla el tamaño del texto. El valor puede definirse en px, em, rem, o %.

Ejemplo:

```
p {  
  
    font-size: 16px; /* Texto de 16 píxeles */  
  
}
```

width y height: Definen el ancho y la altura de los elementos. También se pueden usar diferentes unidades como px, %, vw, y vh.

Ejemplo:

```
.contenedor {  
  
    width: 100%; /* Ancho completo del contenedor padre */  
  
    height: 50vh; /* La mitad de la altura del viewport */  
  
}
```


5.3 Unidades de medida en CSS

Las unidades de medida son esenciales para definir tamaños en CSS. Hay varios tipos de unidades que podemos usar según la situación:

Unidades absolutas: Son fijas y no cambian según el tamaño del viewport. Ejemplos: px (píxeles), cm (centímetros), mm (milímetros), in (pulgadas), pt (puntos), pc (picas).

Unidades relativas: Dependen del entorno y son más flexibles, adaptándose a diferentes tamaños de pantalla. Ejemplos:

em: Relativa al tamaño de la fuente del elemento padre. Si el padre tiene font-size: 16px, 1em es 16px.

rem: Relativa al tamaño de la fuente raíz (<html>).

%: Relativa al tamaño del contenedor padre.

Ejemplo:

```
body {  
    font-size: 16px;  
}  
  
h1 {  
    font-size: 2em; /* 32px, porque 2 * 16px = 32px */  
}
```

Unidades flexibles (vw y vh): Estas unidades son proporcionales al tamaño del viewport.

1vw: 1% del ancho del viewport.

1vh: 1% de la altura del viewport.

Ejemplo:

```
.full-width {  
    width: 100vw; /* Ocupa todo el ancho de la pantalla */  
}
```

Ejercicio práctico #1:

Incorporar CSS externo

Objetivo: Crear un archivo `styles.css`, importarlo en tu HTML, y aplicar estilos al header y footer.

Paso a paso:

1. Crear el archivo `styles.css`: Crea un archivo llamado `styles.css` en la misma carpeta que tu archivo HTML.

Importar `styles.css` en tu HTML: Dentro del `<head>` de tu archivo HTML, agrega la siguiente línea:

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

2. **Agregar contenido en el HTML:** asegúrate de tener un header y un footer en tu archivo HTML.

```
<header>
```

```
    <h1>Mi Sitio Web</h1>
```

```
</header>
```

```
<footer>
```

```
    <p>© 2024 Mi Sitio Web. Todos los derechos reservados.</p>
```

```
</footer>
```

3. Aplicar estilos en el archivo `styles.css`: Estilizar el header y el footer, aplicando un color de fondo y cambiando el estilo de la fuente con las propiedades vistas.

Ejercicio práctico #2:

Modificar la barra de navegación (Navbar)

Objetivo: cambiar el color de los enlaces y el fondo del menú en la lista de navegación.

Paso a paso:

1. Asegurate de tener una lista de navegación como esta:

```
<ul class="nav-list">

  <li><a href="#">Inicio</a></li>

  <li><a href="#">Acerca de</a></li>

  <li><a href="#">Servicios</a></li>

  <li><a href="#">Contacto</a></li>

</ul>
```

2. Aplica los siguientes estilos en `styles.css`:

- Utilizá la propiedad **"list-style"** y el valor **"none"** para quitarle la viñeta a la lista desordenada.
- Utilizá la propiedad **"text-decoration"** y el valor **"none"** para quitarle la apariencia de enlace al anchor dentro de la lista. Utiliza la propiedad `color` para modificar el color del texto dentro del enlace.

3. Este estilo le dará a tu menú de navegación un aspecto más agradable y profesional, mejorando no sólo el diseño y la accesibilidad sino la experiencia de quienes visiten tu sitio web.



Buenos Aires
aprende
Agencia de Habilidades para el Futuro

BA Buenos
Aires
Ciudad