UTN FACULTAD REGIONAL CÓRDOBA
INGENIERÍA EN SISTEMAS DE INFORMACIÓN

Desarrollo de Software
Notas de Clase

Apunte 04 - Hojas de estilos (CSS)

¿Qué es CSS? (*)

CSS o lenguaje de hojas de estilo en cascada (Cascade Style Sheet) se usa para **estilizar elementos** escritos en un lenguaje de marcado como HTML.

CSS separa el contenido de la representación visual del sitio. La diferencia entre un sitio web que implementa CSS y uno que no, es enorme y definitivamente se nota. Antes de CSS, todo el estilo debía incluirse en el marcado HTML. Esto significa que había que describir por separado todos los fondos, los colores de fuente, las alineaciones, etc.

CSS permite estilizar todo en un archivo diferente, creando el estilo allí y después integrando el archivo CSS sobre el marcado HTML. Esto hace que el marcado HTML sea mucho más limpio y fácil de mantener.En resumen, con CSS no tienes que describir repetidamente cómo se ven los elementos individuales. Esto ahorra tiempo, hace el código más corto y menos propenso a errores.

CSS permite tener múltiples estilos en una página HTML, y esto hace que las posibilidades de personalización sean casi infinitas.

Continuando con la analogía utilizada con HTML, la imagen anterior muestra el resultado final del proyecto de casa, ya con las terminaciones correspondientes (estilos). Para lograr estos detalles finales sobre el HTML, CSS utiliza de manera selectiva a sus elementos mediante **reglas**.

Reglas de estilo (*)

Una regla de estilos es un conjunto de instrucciones que define cómo se debe presentar un elemento HTML en una página web. Consiste en un selector que especifica qué elementos HTML serán afectados por la regla, seguido por un bloque de declaración que describe los estilos que se aplicarán a esos elementos seleccionados.

Por ejemplo, la siguiente regla de estilos establece que todos los elementos `<p>` (párrafos) en el documento tendrán un color de texto azul:

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

En esta regla:

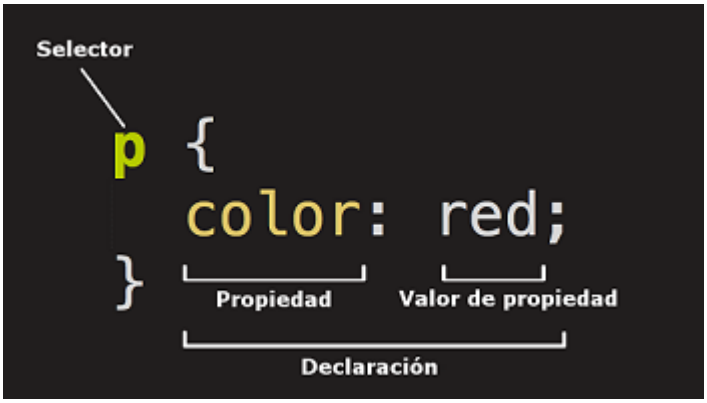
- El selector es p, lo que significa que la regla se aplicará a todos los elementos

  en el documento.

- El bloque de declaración {} contiene la propiedad color y su valor blue, lo que indica que el color del texto de los párrafos será azul.

Las reglas de estilos CSS permiten controlar una amplia variedad de aspectos visuales, como colores, fuentes, márgenes, tamaños y posiciones, entre otros, lo que facilita la presentación y personalización de una página web.

Graficamente la sintaxis de una regla CSS queda representada por las siguientes imágenes:



o también:

Tipos de Selectores

La siguiente tabla resume los diferentes tipos de selectores con su propósito y un ejemplo de aplicación:

Tipo de Selector	**Propósito**	**Ejemplo**
Selector de Elemento	Selecciona todos los elementos de un tipo específico.	`p { color: blue; }` selecciona todos los elementos `<p>` y les aplica un color azul al texto.
Selector de Clase	Selecciona elementos que tienen un valor específico en su atributo `class`.	`.destacado { font-weight: bold; }` selecciona todos los elementos que tienen la clase "destacado" y les aplica negrita.
Selector de ID	Selecciona un elemento específico basado en su atributo `id`.	`#encabezado { font-size: 24px; }` selecciona el elemento con el `id` "encabezado" y le aplica un tamaño de fuente de 24 píxeles.
Selector de Atributo	Selecciona elementos que tienen un atributo específico y opcionalmente un valor específico para ese atributo.	`input[type="text"] { background-color: #f0f0f0; }` selecciona todos los elementos `<input>` de tipo "text" y les aplica un color de fondo gris claro.
Selector Universal	Selecciona todos los elementos en un documento.	`* { margin: 0; padding: 0; }` aplica márgenes y relleno cero a todos los elementos.
Selector de Pseudo-Clase	Selecciona elementos en función de un estado específico o relación con otros elementos.	`a:hover { color: red; }` selecciona todos los enlaces `<a>` cuando el cursor está sobre ellos y les aplica un color rojo al texto.

Tipo de Selector	Propósito	Ejemplo
Selector de Pseudo-Elemento	Selecciona partes específicas de un elemento que no están presentes en el árbol de documentos.	`p::first-line { font-weight: bold; }` selecciona la primera línea de todos los elementos `<p>` y les aplica negrita.

> Otro tipo de selector comúnmente utilizado es el **selector descendiente**. Los selectores descendientes tienen más prioridad sobre otros según el elemento desde el se parte (según su ascendente). En el ejemplo mostrado se aplican los estilos a cualquier enlace **a** que esté contenido dentro de un **div**, por lo que no se indica si es un div class="hijo", class="padre" o id="abuelo". Como existen tres div por encima del enlace, se podría crear un selector "div div div a" que aplicaría a todos los enlaces que tengan tres div ascendentes, y estos estilos tendrían más prioridad que los del selector "div a" del ejemplo anterior. Regla de ejemplo
>
> **div a{background-color:#009cde;color:#ffffff;}**
>
> **div div div a{background-color:red;color:yellow;}**

> Cabe mencionar que también es posible combinar elementos hermanos mediante el operador **'~'** o hermanos adyancentes mediante **'+'**; o hijos directos mediante A **'>'** B.

¿Dónde escribir reglas CSS? (*)

Se pueden escribir reglas CSS en:

1. Dentro de un **archivo externo** con extensión .css:
    - Crear un nuevo archivo en una carpeta styles (por convención) y copiar las líneas de código.
    - Luego abrir el archivo html pegar la siguiente línea dentro del :

```
<link  href="styles/style.css"  rel="stylesheet"  type="text/css">
```

    - Guardar el archivo .html y recargar la página para ver los estilos aplicados
2. Dentro de `<head>` agregar el **elemento** `<style>`:

    Dentro del bloque `<head$>` agrega la etiqueta `<style$>`. Por ejemplo:

```
<head>
    <style>
        p: red;
    </style>
</head>
```

Con lo que todo texto encerrado en una etiqueta `<p>` se mostrará de color rojo.

3. **En línea**:

   Directamente sobre un elemento HTML agregar el atributo style. Por ejemplo:

   ```
   <p  style="color: blue; text-align: center;">
   ```

   Esto causaría un efecto similar al ejemplo anterior, pero solo para el texto del párrafo correspondiente, dejándolo de color azul.

Especificidad de selectores

La especificidad de un selector determina qué estilo se aplicará si varios estilos entran en conflicto. Se calcula mediante el número de ID, clases y elementos en un selector. Un selector más específico prevalecerá sobre uno menos específico.

Cuando dos reglas tienen la misma especificidad, la regla que aparece más tarde en la hoja de estilos se aplicará. Esto se conoce como "última regla en cascada".

Por ejemplo, dado el siguiente código HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de CSS en cascada</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div id="container">
    <p class="text">Este es un párrafo de ejemplo.</p>
  </div>
</body>
</html>
```

Si se aplican las siguientes reglas CSS:

```
/* Regla más específica */
#container p.text {
  color: blue;
}

/* Regla menos específica */
.text {
  color: red;
}
```

- La regla #container p.text es más específica ya que contiene un ID (#container) y una clase (text), mientras que la regla .text solo tiene una clase.
- Ambas reglas se aplican al mismo elemento `<p class="text">`, pero el color definido en la regla más específica (blue) prevalecerá sobre el color definido en la regla menos específica (red).

Ahora supongamos que se agrega una regla extra en línea, es decir en el propio HTML:

```
<!DOCTYPE html>
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Ejemplo de CSS en cascada</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <div id="container">
    <p class="text" style="color: green;">Este es un párrafo de ejemplo.</p>
  </div>
</body>
</html>
```

En este ejemplo, se ha añadido un estilo en línea al párrafo con la clase text que establece el color del texto en verde. Ahora, la jerarquía de prioridad sería la siguiente:

1. Estilo en línea (mayor prioridad)
2. Reglas CSS en el documento HTML
3. Hoja de estilos externa (styles.css)

> El color del texto del párrafo se establecerá como verde debido al estilo en línea, ignorando completamente las reglas de CSS definidas en el archivo externo styles.css.

Variables

Los sitios web complejos tienen una gran cantidad de CSS, a menudo con muchos valores repetidos. Por ejemplo, el mismo color puede usarse en cientos de lugares diferentes, lo que requiere una búsqueda global y reemplazo si ese color necesita cambiar. Las **propiedades personalizadas** o simplemente **variables CSS** permiten que un valor se almacene en un lugar y luego se haga referencia en muchos otros lugares. Un beneficio adicional son los identificadores semánticos. Por ejemplo, "--main-text-color" es más fácil de entender que #00ff00, especialmente si este mismo color también se usa en otros contextos.

> Las propiedades personalizadas están sujetas a la cascada y heredan su valor de su padre.

Regla `!important`

El uso de `!important` es una forma de anular las reglas de cascada normales y forzar que una regla específica tenga prioridad sobre otras reglas, incluso si es menos específica o se declara en un lugar diferente en la hoja

de estilos. Esto puede ser útil en situaciones donde necesitas asegurarte de que un estilo específico prevalezca.

¿Cómo crear variables CSS?

- Para crear una variable: "--nombre-variable: unValor"
- Para usar una variable declarada en un ámbito: "var(--nombre-variable)"
- También podemos definir un valor de respaldo, por ejemplo:

```
html{
--color-defecto: #000
}

.mi-clase{
    border-right: 1px  solid  var(--defecto, black);
}
```

Ejemplo de Aplicación

Como práctica inicial de CSS se propone crear una página y darle los estilos necesarios para que el resultado sea lo más similar a la imagen que se muestra a continuación:

Bibliografía o Referencia

- MDN References - CSS