



HTML

Orígenes e importancia de los estilos CSS

El CSS que escribes no es el único que afecta a una página web. Existen diferentes orígenes de los estilos CSS que se aplican:

- **Estilos Predeterminados del Navegador:** Estos son los estilos que el navegador utiliza para mostrar los elementos HTML. Los estilos predeterminados varían ligeramente entre los navegadores, pero en general establecen propiedades como espaciado en los títulos y párrafos, espaciado y estilo de las listas, colores de los enlaces y tamaños de letra por defecto.
- **Extensiones del Navegador y Sistema Operativo:** Algunas extensiones del navegador o incluso el sistema operativo pueden agregar estilos CSS adicionales. Estos estilos pueden incluir modificaciones en propiedades como el tamaño de fuente. Tienen una prioridad mayor que los estilos predeterminados del navegador.
- **CSS de Autor:** Este es el CSS que tú mismo escribes para personalizar la apariencia de tu página web. Los estilos de autor se aplican mediante una hoja de estilos que creas y en la que defines reglas CSS para los elementos de tu página. Estos estilos tienen una prioridad mayor que los estilos predeterminados del navegador y las extensiones del navegador.
- **Importancia en Declaraciones CSS:** En el caso de que haya declaraciones conflictivas en tu CSS de autor, puedes utilizar la palabra clave `!important` para indicar que un estilo específico debe tener mayor prioridad sobre otros estilos. Esto otorga una importancia adicional a las declaraciones con `!important` dentro de tu CSS de autor.
- **Importancia a Nivel de Usuario y Navegador:** A nivel de usuario, los estilos definidos en el propio navegador por el usuario tienen una prioridad mayor que los estilos de autor. Asimismo, los estilos definidos a nivel del navegador tienen una prioridad mayor que los estilos de autor. Estos estilos a nivel de usuario y navegador pueden afectar propiedades como el tamaño de fuente o los estilos de los enlaces.

Orden de prioridad de los orígenes

El orden de prioridad de estos orígenes, de menor a mayor, es el siguiente:

1. Estilos predeterminados del navegador.
2. Estilos del usuario (extensiones del navegador y sistema operativo).
3. CSS de autor.
4. CSS de autor con la palabra clave `!important`.
5. `!important` a nivel del usuario.
6. `!important` a nivel del navegador.

Los estilos predeterminados varían ligeramente de un navegador a otro, pero en general hacen lo mismo. Para evitar problemas con diferentes navegadores, a menudo se usan hojas de estilo llamadas reset, una muy popular es normalize, puedes [descargarla aquí](#). Aunque también puedes crear tu propio reset. Estos estilos establecen una base para las propiedades que se tratan de forma diferente en los distintos navegadores.

Ejemplo de sobrescribir estilos del navegador

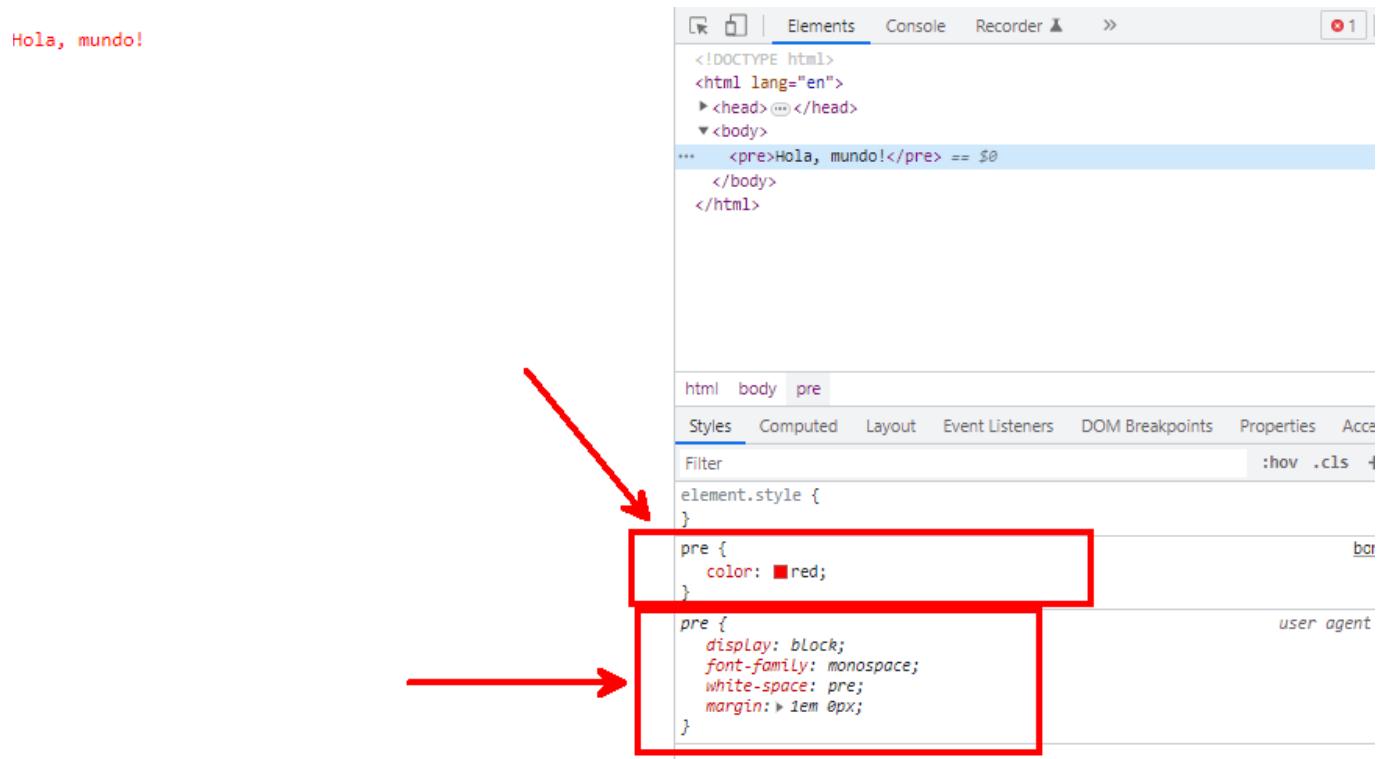
En este ejemplo, vamos a mostrar cómo sobrescribir los estilos del navegador para un elemento específico:

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <link rel="stylesheet" href="style.css" />
  </head>
  <body>
    <pre>Hola, mundo!</pre>
  </body>
</html>
```

```
// style.css

pre {
  color: red;
  display: inline;
}
```

En el código anterior, se establece el color rojo para la etiqueta `<pre>`, si nos vamos a las developer tools del navegador podemos observar que es la única regla de autor que está aplicando y el resto son reglas que por defecto añade el navegador.



Ahora queremos sobrescribir la propiedad `display: block` y aplicar nuestro propio estilo. Para ello, podemos usar la propiedad `display` con el valor `inline`:

```
// style.css

pre {
  color: red;
  display: inline;
}
```

Al añadir esta regla en nuestra hoja de estilos, sobrescribimos el estilo del navegador y aplicamos el nuevo estilo deseado.

The screenshot shows the Chrome DevTools interface. At the top, the 'Elements' tab is selected. Below it, the DOM tree shows a single node: a pre element with the text 'Hola, mundo!'. In the bottom right corner of the DOM tree, there is a red arrow pointing to the 'user agent' style block for the pre element.

```

<!DOCTYPE html>
<html lang="en">
  <head>...</head>
  <body>
    ... <pre>Hola, mundo!</pre> == $0
  </body>
</html>

```

The 'Styles' tab is selected in the bottom navigation bar. It lists two style blocks:

- element.style {**
- pre {** (highlighted with a red box)

The 'pre' style block contains the following rules:

```

color: red;
display: inline;
}

```

The 'user agent' style block for 'pre' contains the following rules:

```

display: block;
font-family: monospace;
white-space: pre;
margin: 1em 0px;
}

```

Reglas de CSS en orden de importancia

A continuación, se muestra el orden de importancia de las reglas CSS, de menor a mayor:

1. Reglas comunes: Propiedades como `font-size`, `background` o `color`.
2. Reglas con la propiedad `animation`.
3. Reglas con la palabra clave `!important`, en el mismo orden en que aparecen en la hoja de estilos.
4. Reglas con la propiedad `transition`.

Es importante tener en cuenta que las reglas con las propiedades `animation` y `transition` tienen mayor importancia que las reglas comunes y las reglas con la palabra clave `!important`.

Consejos adicionales

Aquí hay algunos consejos adicionales para trabajar con la cascada en CSS:

- Evita utilizar ID en tus selectores, ya que aumentan la especificidad y pueden complicar la cascada.
- No uses la palabra clave `!important` a menos que sea necesario. Es mejor evitar su uso debido a la dificultad de anularla.

Recuerda que los navegadores utilizan un algoritmo llamado cascada para decidir qué reglas aplicar a cada elemento cuando hay conflictos. La cascada significa que los estilos que vienen después anulan los que estaban antes y es común que la especificidad rompa esta cascada. Esto es uno de los principales objetivos de la cascada que es la resolución de conflictos y que hay otros elementos importantes junto a la cascada que es la especificidad y la herencia que controlan qué estilos se aplican a un elemento.