

ArgentinaPrograma
YoProgramo

CSS

por Leonardo Blautzik, Federico Gasior y Lucas Videla

Julio / Diciembre 2021



Ministerio de
Desarrollo Productivo
Argentina

CSS (Cascading Style Sheets)

Es un lenguaje de hojas de estilo utilizado para describir la presentación de los elementos de un documento escrito en un lenguaje de marcado (como por ejemplo HTML).

Esta diseñado para trabajar con múltiples selectores que aplican reglas a un conjunto de elementos, y que al poder ser un archivo aparte, puede definir la estructura entera de un sitio o usarse de forma modular en donde se necesite

La C de cascada es por la forma de aplicar sus reglas, ya que si existe más de una regla del mismo tipo que apunte al mismo elemento, se aplicará un comportamiento ordenado y predecible

Sintaxis

Selector del elemento <h1>, que posee una lista de declaraciones como clave-valor. CSS es sensitivo a mayúsculas y minúsculas

```
h1 {  
    color: green;  
    letter-spacing: 10px;  
}
```

```
h2 {  
    color: blue;  
    font-size: 20px;  
}
```

Los espacios extras son ignorados. Puede ser escrito en una línea

Especificaciones

CSS, al igual que HTML, dispone de especificaciones dentro de W3C que definen las características que tiene un documento de CSS

A partir de CSS3, el mismo se modularizó y las versiones y avances dependen de cada módulo y no tiene una versión global por si misma

A pesar de todas las nuevas características que va incluyendo en sus nuevas versiones, CSS respeta las definiciones originales, por lo tanto sitios ya antiguos, se deberían ver igual que en su momento (cosa que por ejemplo no ocurre en HTML)

Compatibilidad

A pesar de las especificaciones definidas en las versiones de CSS; cada navegador o interprete de CSS puede no comprender todas las características (sobre todo las más nuevas), o tener incluidas características experimentales

Si tienen dudas a la hora de aplicar alguna característica que pueda ser relativamente moderna, pueden consultar el sitio CAN I USE donde cada una de estas características tiene detallado su soporte para la mayoría de navegadores actuales. Los interpretes suelen especificarlo en su propia documentación

Ejemplo para letter-spacing: <https://caniuse.com/?search=letter-spacing>

Formas de incluir CSS en un documento

- En un archivo aparte
- En el mismo documento HTML
- Directamente en el elemento

En un archivo aparte

Se pueden incluir uno o varios elementos `<link>` en el `<head>` que lo que van a hacer es incluir e implementar estas hojas de especificaciones al documento. Lo recomendado es crear una por cada tipo de modulo a trabajar, por ejemplo colores, títulos, tablas, etc

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

En el mismo documento HTML

En muchos casos puede ocurrir que algunos estilos son muy específicos de un solo documento y no vale la pena crear un archivo aparte para este fin. A través del elemento `<style>` que debe ubicarse dentro de `<head>`

```
<style type="text/css">  
/* Estilos */  
</style>
```


Directamente en el elemento

Para casos específicos, donde solo un elemento, una única vez va a ser modificado con algún estilo en particular se puede incluir los mismos directamente como un atributo de HTML: el “style”

```
<h1 style="color: green; letter-spacing: 10px;">Hola mundo</h1>
```

Aplicando estilos

Como vimos anteriormente, podemos aplicar estilos a elementos HTML poniendo su nombre de etiqueta y dentro las reglas a aplicar. Si quiero aplicar las mismas reglas a varios elementos, puedo separarlos con una coma

```
h1, h2, h3, h4, h5 {  
    font-size: 12px;  
}
```

Este ejemplo además demuestra que puedo sobrescribir el comportamiento por defecto de los elementos, tales como el tamaño en los distintos títulos. Esto no es excusa para no utilizar los elementos HTML adecuados

Aplicando un estilo a varios elementos distintos: Clase

Va a haber pocas veces donde me interese cambiar el estilo de un elemento entero. Normalmente me interesa agregarle un estilo a un conjunto de datos que sean similares en algún sentido, como responsabilidad, contexto, etc

Para ello se puede definir un atributo en los elementos HTML el cual es la clase `class`. De esta manera, desde CSS podemos referenciarlos utilizando un `.` delante del nombre

Clase: ejemplo

```
<p>Hola</p>
```

```
<p class="destacado">Párrafo importante</p>
```

```
<p>Otra cosa</p>
```

```
<p class="destacado">Otro importante</p>
```

```
.destacado {  
    color: black;  
    font-weight: bold;  
}
```

Aplicando un estilo a un único elemento: Id

Hay casos donde un elemento trasciende del resto y debe ser tratado como un elemento único. Para esos casos podemos definir el atributo `id` en cualquier elemento HTML, al igual que `class`, pero solo puede existir un atributo `id` con un valor para cada elemento deseado. No pueden repetirse o pueden causar comportamientos inesperados

```
<h1 id="encabezado-principal">¡Bienvenidos!</h1>
```

```
#encabezado-principal {  
    font-size: 30px;  
    margin-bottom: 10px;  
}
```

Combinando selectores

Todas las reglas vistas y por verse, pueden ponerse juntas implicando que el elemento debe cumplir con todas ellas. A la vez sirve para aplicar reglas más genéricas y más específicas, dependiendo el caso

```
.destacado {  
    font-weight: bold;  
}  
p.destacado {  
    color: black;  
}  
article.destacado {  
    color: red;  
}
```

Selectores dependiendo de la ubicación de otros elementos

```
/* Artículos dentro de una sección (hijos, nietos, etc) */  
section article {  
    font-size: 12px;  
}  
  
/* Artículos únicamente hijos de una sección */  
section > article {  
    font-size: 12px;  
}  
  
/* Reglas aplicadas al p adjacente (a posterior) a un h1 */  
h1 + p {  
    font-size: 200%;  
}
```

Selectores basados en estado

```
a:link {  
    color: red;  
}  
a:visited {  
    color: white;  
}  
a:hover {  
    text-decoration: none; /* Elimina el subrayado */  
}
```


Herencia

Algunas propiedades de CSS declaradas en elementos padres son heredadas por sus hijos. Un ejemplo de estas propiedades puede ser color. Si yo defino un color para `<main>`, todos los elementos hijos tendrán ese color, salvo que decidan modificar esa propiedad.

Existen ciertos valores que se pueden designar a este tipo de propiedades para modificar su comportamiento de herencia:

- `inherit`: Activa la herencia (valor por defecto de estas propiedades)
- `initial`: Deja el valor de la propiedad como la inicial, rompiendo la herencia
- `unset`: Reinicia la propiedad, para que funcione de manera por defecto (es decir, `inherit` para estas propiedades, `initial` para las otras)

Conflictos (y resolución de cascada)

Cuando dos selectores son completamente iguales y modifican la misma regla, se aplicará el orden de cascada. Al aplicarlo, se determina que la última declarada será la utilizada en el documento

En el caso ejemplo, el color que van a presentar los `<p>` va a ser blue o azul

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

En el caso de que dos selectores no sean iguales pero tengan elementos coincidentes y modifiquen la misma regla, la regla que finalmente termine aplicando se calculara con un calculo de especificidad, que da un puntaje según los tipos de selectores que se utilicen

Calculadora: <https://specificity.keegan.st/>

¡Importante!

Si quisiera que una regla se aplique siempre a pesar de los conflictos o reglas de especificidad, puedo declararlo como importante. De esta manera, tendrá más prioridad que las otras reglas.

Si existen múltiples reglas con importante, se realizarán los cálculos como los vistos anteriormente pero solo entre los importantes

```
.encabezado {  
    color: red;  
}  
h1 {  
    color: black !important;  
}
```

Unidades y valores

Cada propiedad en CSS tiene un conjunto de valores definidos que están permitidos. Muchos de estos son literales o palabras claves, como “inherit”, mientras que otros son más genéricos refiriéndose a una unidad, como un color o un largo

En css los colores están definidos con un tamaño de 24 bits, lo que permite obtener casi 17 millones de colores (2^{24}). Las formas de representarlo en CSS son:

- Palabra clave: Un listado definido de colores en inglés (como: red, blue, green, black, white)
- RGB y RGBA: Sistema expresado en cantidad de color rojo, verde y azul, entre 0 a 255. El ultimo valor A de opacidad es opcional, y regula entre 0 y 1 que tan opaco será el color
- HEX RGA: Similar al anterior pero expresado en hexadecimal
- HSL: Sistema de colores pensado para diseñadores, donde el primer valor es el HUE o la base del color, en un ángulo de un círculo, y los otros dos son la saturación o el contraste y la luminosidad o el brillo

En css los números se dividen en subgrupos los cuales son:

- integer: Valor entero
- number: Valor real o decimal
- dimension: Es un número compuesto por una unidad. Por ejemplo grados (deg), segundos (s) o pixeles (px).
- porcentaje: Fracción de otro valor, normalmente del contenedor

Estilos de tabla

Las tablas disponen de estilos especiales para aprovechar todas sus funcionalidades. Pero colocar un estilo solo en la tabla nos va a dejar a mitad de camino de lo que probablemente deseemos, que es generar estilo para toda la tabla

```
table {  
    table-layout: fixed; /* Tamaño para celdas según primer fila */  
    width: 100%;  
    border-collapse: collapse; /* Quita la separación entre celdas */  
    border: 2px solid #000000; /* Borde para tablas */  
    background: linear-gradient(to bottom, #eedddd, #ddcccc);  
}
```


Estilos de tabla: hijos

Para aplicar propiedades específicas a las celdas, debemos pedirselo a cada elemento que necesitemos

```
thead th:nth-child(1) { width: 60%; }
```

```
thead th:nth-child(2) { width: 40%; }
```

```
th, td {  
  border: 2px solid black;  
  padding: 5px 3px;  
}
```

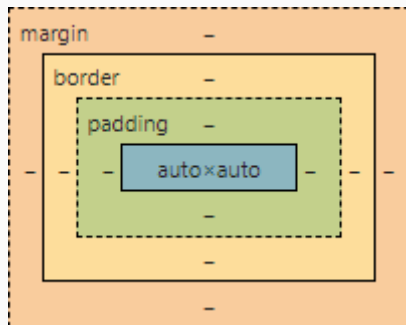
Estilos de tabla: Rayas de cebra

```
tbody tr:nth-child(odd) > * {  
    background-color: rgba(0, 0, 0, 0.15);  
}
```

```
tbody tr:nth-child(even) > * {  
    background-color: rgba(0, 0, 0, 0.025);  
}
```

Modelo de caja

Todos los elementos en CSS disponen de una caja que los rodea. Puede ser principalmente de 2 tipos: de bloque o en línea (block o inline). Cada caja tiene propiedades como ancho (width), alto (height), relleno (padding), borde (border) y margen (margin). El tipo puede ser reemplazado a través de la regla `display`



Cajas bloque

- La caja realizará un salto de línea
- La caja llenará el espacio disponible por su contenedor
- Las propiedades de ancho y alto son respetadas
- Las propiedades de relleno, borde y margen empujarán a los otros elementos lejos de la caja

Algunos elementos HTML que tienen este tipo por defecto son: `<h1>...<h6>`, `<p>`, `<div>`

- La caja NO realizará un salto de linea
- El ancho y alto no aplicarán
- Las propiedades de relleno, borde y margen aplicarán pero solo empujarán a los otros elementos lejos de la caja en el eje horizontal

Algunos elementos HTML que tienen este tipo por defecto son: `<a>`, ``, ``

Inspeccionar el código CSS

Al igual que en HTML, a través de las herramientas de desarrollador podemos observar que estilos están aplicados a cada elemento del DOM, y a la vez modificarlos en tiempo real.

En la mayoría de navegadores, las herramientas de desarrollador también disponen de otras herramientas para ayudarnos a entender y trabajar con CSS

Al igual que en HTML, podemos validar nuestro código en el sitio de W3

<https://jigsaw.w3.org/css-validator/>

CSS

¡Muchas Gracias!

continuará...



Ministerio de
Desarrollo Productivo
Argentina