

# 305 CSS - Reglas y selectores

## Reglas CSS

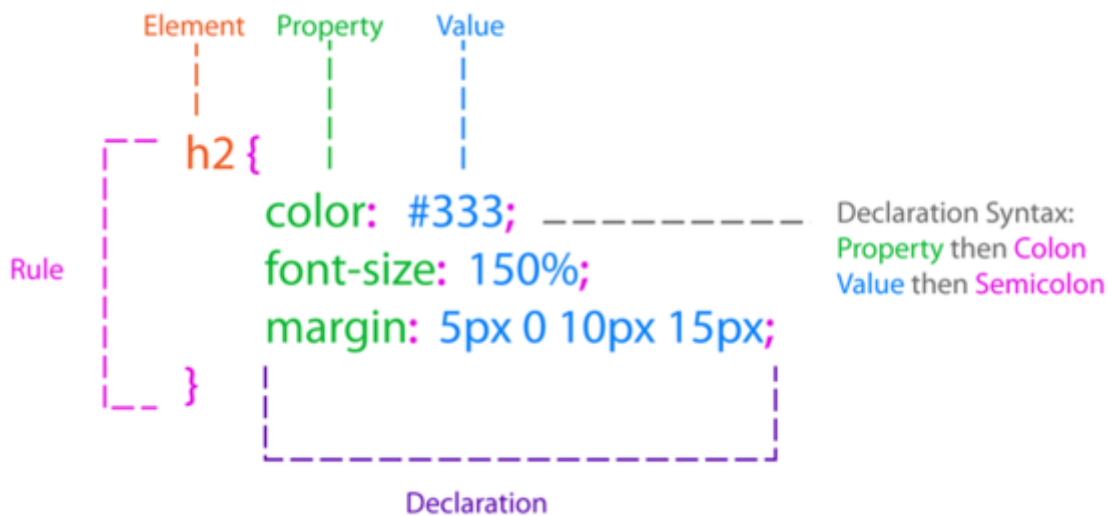
Un conjunto de reglas CSS se compone de un selector y un bloque de declaración:

```
selector(s) {  
    property1: value1;  
    property2: value2;  
}
```

Visualicémoslo:



- **selector:** apunta al elemento(s) al que aplicarás la regla de estilo,
- **bloque de declaración:** contiene **una o más declaraciones *separadas por punto y coma***. Estas declaraciones especifican las propiedades del elemento y los valores que desea establecer.



Ahora, si vuelves al ejemplo anterior, puedes leer las reglas CSS que te hemos proporcionado:

```
h1 {  
    color: pink;  
    text-decoration: underline;  
}
```

Nos dirigimos a la etiqueta `<h1>` y establecemos su *propiedad* `color` (se refiere al color de la fuente) al *valor rosa*, y su *propiedad* `text-decoration` al *valor underline*.

Sin embargo, detengámonos aquí un momento. ¿Qué pasa si tenemos más de una etiqueta `<h1>` en nuestra página? ¿Tendrán todas el mismo estilo? Sí. Pero, ¿siempre queremos eso? Bueno... en realidad no. Así que aquí está cómo podemos manipular qué elementos deben obtener estilos y cuáles no.

## CSS (selectores)

### Selectores de tipo

[Los selectores de tipo](https://developer.mozilla.org/en-US/docs/Web/CSS/Type_selector) emparejar elementos por su nombre (de nodo). En HTML, los elementos son las etique[https://developer.mozilla.org/en-US/docs/Web/CSS/Type\\_selector](https://developer.mozilla.org/en-US/docs/Web/CSS/Type_selector)tas que añadimos, por ejemplo `<p>`, `<header>` o `<div>`. La estructura de los selectores de tipo es:

```
tag-name {  
    rule: value;  
}
```

Ya ha utilizado este selector en el ejemplo anterior. Ahora, vamos a entender lo que está pasando.

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

Este trozo de código le dice al documento HTML que establezca el color del texto dentro de todas las etiquetas `<p>` en azul.

El selector de elemento **afectará a todos los elementos de su clase, sin importar cuántos haya.**

Este es el ejemplo.

```
<p>First paragraph</p>  
<p>Second paragraph</p>  
<p>Third paragraph</p>
```

```
p {  
    color: navy  
}
```

**Hora de practicar:**

Modifica el CodePen anterior creando:

- Dos etiquetas `<h4>`. El color del texto será `darkred`.
- Envuelve la palabra second con un elemento `<span>`. Esta palabra debe ser `"yellow"`.

## Selectores de clase

En HTML y CSS, puedes identificar un grupo de elementos HTML con las mismas características o estilos utilizando un [atributo de clase](#).

La clase es un atributo del elemento(s) que está intentando cambiar, y **requiere un nombre**. La estructura del selector de clase es:

```
/* en el archivo CSS */  
.class-name {  
    rule: value;  
}
```

Ejemplo:

```
/* fíjate en el punto ( . ) antes de la palabra red */
.red {
    background-color: red;
}
```

Convenciones para nombrar clases:

- Debes nombrar las clases utilizando sólo **letras minúsculas**.
- Si quieres que el nombre de una clase esté compuesto por **varias palabras**, por favor **separalas usando guiones -**, por ejemplo: clase con varios nombres
- No puede utilizar espacios en el nombre de una clase\*\*.

```
/* MAL: */
.cool class {
    color: purple;
}

/* BIEN */
.cool-class {
    color: purple;
}
```

Los [selectores de clase](#) coinciden con todos los elementos basándose en el valor del atributo `class` de un elemento.

El ejemplo - haga clic en la pestaña CSS para ver las reglas CSS:

```
<div class="red-box">
  puede estilizar esta caja con el selector .red-box
</div>

<div class="blue-box">
  puede estilizar esta caja con el selector .blue-box
</div>

<div class="red-box">
  puede estilizar esta caja con el selector .red-box
</div>
```

```
.red-box {
    background-color: red;
}

.blue-box {
```

```
background-color: blue;
}
```

Estos selectores no hacen nada por sí mismos. Es sólo una forma de agrupar los elementos en una categoría de "red-box" o "blue-box".

### Hora de practicar:

- Crea un elemento `blue-box` más.
- Crea un elemento `yellow-box` para convertir el color en amarillo. Será una etiqueta `<p>` con el texto *'This is a yellow color text'*.

¡Recuerda añadir un punto antes del nombre de la clase en tu archivo CSS!

## Analogía

Imagina que estás en un campamento de verano y, de repente, los monitores dividen a los campistas en dos equipos: rojo y azul. Dividir a los campistas en dos grupos no cambia necesariamente su aspecto, pero entonces imaginemos que el monitor dice:

"Todos los del grupo rojo, pintaos la cara de rojo".

"Todos los del grupo azul: ¡pintaros la cara de azul!".

Ese es un comportamiento diferente. Este comando cambiará tu aspecto y sólo se aplica a tu grupo. Traduzcámoslo a código:

```
.red-group {
  background-color: red;
  width: 200px;
  height: 200px;
}

.blue-group {
  background-color: blue;
  width: 200px;
  height: 200px;
}
```

```
<div class="red-group">
  <li>Madison</li>
  <li>Rashawn</li>
  <li>Rashad</li>
  <li>Rodger</li>
  <li>Hugh</li>
```

```

    <li>Joey</li>
</div>

<div class="blue-group">
    <li>Ethyl</li>
    <li>Kayleigh</li>
    <li>Jayson</li>
    <li>Jasen</li>
    <li>Gilberto</li>
    <li>Marley</li>
</div>

```

En este caso, el selector es el **class**. Significa que seleccionarás todos los elementos con la clase `grupo-rojo`. Ahora, todas las propiedades listadas dentro de las llaves se aplicarán a los elementos con la clase `grupo-rojo`.

## Hora de practicar:

Crea un archivo HTML con la siguiente estructura:

```

<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <title>How to use CSS classes - practice</title>
  </head>
  <body>
    </body>
</html>

```

- Haz que `#00D1AE` sea el color de fondo de tu etiqueta body.
- Crea un `<div>` con la clase `menu`, y asígnale los siguientes atributos:
  - `width: 400px`.
  - `height: 50px`.
  - `background-color: #00D1AE`.
- Crea otro `<div>` con la clase `content`, y asígnale los siguientes atributos:
  - `width: 400px`.
  - `height: 200px`.
  - `background-color: red`.
- Añade un `<div>` más y haz que parezca `menu`.
- Añade un último `<div>` y haz que parezca `content`.

## Clases múltiples

Un elemento HTML puede tener **múltiples** clases adjuntas. Si desea añadir varias clases a un elemento, incluya cada clase separada por un espacio:

```
<div class="red-box dimensions">
    ...
</div>
```

Lo siguiente es un error común que no funcionará:

```
<div class="red-box" class="dimensions">
    ...
</div>
```

**Hora de practicar:**

¿Recuerdas las clases `menu` y `content`? Tienen el mismo ancho, ¿verdad?

Crea una clase llamada *important*. Esta clase tendrá la regla CSS:

```
.important {
    font-weight: bold;
}
```

¿Cómo incluirías esta (la segunda) clase en tu HTML?

## Selectores ID

Un [selector ID](#) identifica un **elemento único e irrepetible** en el HTML.

Los selectores ID se identifican por el símbolo **hash (#)**.

La convención de nomenclatura es la misma que para las clases: si tiene **múltiples palabras**, **combínalas con un guión (-)**.

Los elementos **pueden y sólo** deben tener **un ID**.

## Ejemplo guiado

Vamos a reutilizar el mismo ejemplo que hemos creado anteriormente: **html-css** con dos archivos - *index.html* y *style.css*.

## Juguemos

Creemos una etiqueta `div` (dentro de la etiqueta *body* en el documento *index.html*) con un atributo `id`, algo que represente de forma única a ese div en el código, digamos `cover-box`:

```
<!-- index.html -->
<div id="cover-box">
    ...
</div>
```

Si quieres cambiar el fondo de ese elemento, lo seleccionarás añadiendo una regla selectora ID dentro del CSS (indicando un `#` antes del nombre del id).

```
/* style.css */
#cover-box {
    background-color: black;
    color: pink;
    width: 100px;
    height: 50px;
}
```

Ahora crea otro elemento con el mismo ID en tu documento HTML.

```
<!-- index.html -->
<div id="cover-box">
    First cover box
</div>

<div id="cover-box">
    Second cover box
</div>
```

¿Qué ocurre? ¿Cambia todos los elementos con ese id?

Sí, cambiará la apariencia de todos los elementos. Los IDs son únicos, pero Chrome aplicará los mismos estilos a cada elemento con el mismo ID. **Esto no es cierto en todos los navegadores** y puede causar muchos problemas en el futuro.

**¡Asegúrate de que los IDs son únicos en tu página HTML!**

## Selectores universales

El [Selector universal](#) coincide con un único elemento de cualquier tipo. Se identifica por un asterisco (`*`).

Omitir el asterisco con selectores simples tiene el mismo efecto. Por ejemplo

- `*.warning`
- `.warning`



Estos dos selectores se consideran iguales, ya que ambos se dirigirán a todos los elementos que tengan la clase `warning`, independientemente del tipo de elemento (`div` o `p` o `whatever`)

```
* {  
    font-family: monospace;  
}
```

Ten mucho cuidado cuando utilice el selector universal, ya que ralentizará su página considerablemente, después de todo, ¡aplicará ese estilo a TODOS los elementos de su página!

Normalmente, esta es la *primera* regla en su documento CSS para aplicar estilos globales.

## Hora de practicar:

Crea una regla para tu `index.html`:

- Todos los elementos deben tener `color: verde;`.
- Cuando veas el resultado, cambia el estilo de la `cover-box`. El estilo `cover-box` debe tener `color: navy;`. ¿Qué ocurre? ¿Cuál es el color del texto dentro de la etiqueta `div` con el `id="cover-box"`?

El color dentro de la etiqueta `div` con el `id="cover-box"` debe ser `navy`, ya que es el más específico y debe sobrescribir la regla global.

```
* {  
    color: green;  
}  
  
#cover-box {  
    color: navy;  
}
```

Una regla más: **Cada documento (html, css, js, etc.) se lee de arriba a abajo** (línea 1, línea 2, línea 3, etc.) - lo que significa que la última regla que apliques será la que verás en la página.