
2.2 Selectores

Los selectores nos **ayudan a indicar el elemento sobre el que se van a aplicar los estilos**. Los selectores pueden apuntar a elementos específicos, clases, identificadores o incluso atributos de un elemento.

Existen muchos tipos de selectores y algunos de los más destacados son los que se detallan a continuación.

2.2.1 Selector universal

Sintaxis: `* { atributo:valor; }`

Ejemplo: `* { color: grey; }` /* El estilo se aplicará a todos los elementos de la página*/

El **selector universal (*)** es útil cuando deseas **aplicar un estilo a todos los elementos en una página web sin excepción**. Es una forma rápida y sencilla de establecer reglas generales que afectarán a todos los elementos en el documento. Por ejemplo, si deseas eliminar todos los márgenes y rellenos predeterminados de los elementos en tu página para empezar desde cero, puedes usar el selector universal para restablecerlos.

```
* {  
  margin: 0;  
  padding: 0;  
}
```

2.2.2 Selector etiqueta

Sintaxis: `etiqueta { atributo:valor }`

Ejemplo: `p {color: green;}` /* El estilo se aplicará a todos los elementos <p>.*/*

Ejemplo: `h1,h2,h3 {color: black; font-family:Arial, Helvetica;}` /* El estilo se aplicará a todos los encabezados>.*/*

En las hojas de estilo complejas, **es habitual agrupar las propiedades comunes de varios elementos en una única regla CSS y posteriormente definir las propiedades específicas** de esos mismos elementos. El siguiente ejemplo establece en primer lugar las propiedades comunes de los títulos de sección (color y tipo de letra) y a continuación, establece el tamaño de letra de cada uno de ellos:

```
h1, h2, h3 {  
  color: #8A8E27;  
  font-family: Arial, Helvetica, sans-serif;  
}  
h1 { font-size: 2em; }  
h2 { font-size: 1.5em; }  
h3 { font-size: 1.2em; }
```

2.2.3 Selector clase

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en **utilizar el atributo class de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar**:

Sintaxis: `.clase { atributo:valor }`

Ejemplo: `.destacado {color: red;} /* El estilo se aplicará a cualquier elemento que tenga la clase .destacado*/`

La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo class:

```
<body>
  <p class="destacado">Lorem ipsum dolor sit amet...</p>
  <p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a>
  accumsan...</p>
  <p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>
</body>
```

Los selectores de clase **son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos**. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes.

Ejemplo:

```
.aviso {
padding: 0.5em;
border: 1px solid #98be10;
background: #f6feda;
}
.error {
color: #930;
font-weight: bold;
}
<span class="error">...</span>
<div class="aviso">...</div>
```

El elemento `` tiene un atributo `class="error"`, por lo que **se le aplican las reglas CSS indicadas por el selector .error**. Por su parte, el elemento `<div>` tiene un atributo `class="aviso"`, por lo que su estilo es el que definen **las reglas CSS del selector .aviso**.

En ocasiones, es necesario **restringir el alcance del selector de clase**. Si se considera de nuevo el ejemplo anterior:

```
<body>
  <p class="destacado">Lorem ipsum dolor sit amet...</p>
  <p>Nunc sed lacus et <a href="#" class="destacado">est adipiscing</a>accumsan...</p>
  <p>Class aptent taciti <em class="destacado">sociosqu ad</em> litora...</p>
</body>
```

¿Cómo es posible aplicar estilos solamente al párrafo cuyo atributo class sea igual a destacado?
Combinando el selector de tipo y el selector de clase, se obtiene un selector mucho más específico:

```
p.destacado { color: red }
```

El selector **p.destacado** se interpreta como **"aquellos elementos de tipo <p> que dispongan de un atributo class con valor destacado"**.

No debe confundirse el selector de clase con los selectores anteriores:

```
/* Todos los elementos de tipo "p" con atributo class="aviso" */
p.aviso { ... }
/* Todos los elementos con atributo class="aviso" que estén dentro de cualquier elemento de tipo "p" */
p .aviso { ... }
/* Todos los elementos "p" de la página y todos los elementos con atributo class="aviso" de la página */
p, .aviso { ... }
```

Por último, **es posible aplicar los estilos de varias clases CSS sobre un mismo elemento**. La sintaxis es similar, pero los diferentes valores del atributo class se **separan con espacios en blanco**.

Ejemplo:

```
.error { color: red; }
.destacado { font-size: 15px; }
.especial { font-weight: bold; }
<p class="especial destacado error">Párrafo de texto...</p>
```

Al párrafo anterior se le aplican los estilos definidos en las reglas .especial, .destacado y .error, por lo que en el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel.

Si un elemento dispone **de un atributo class con más de un valor, es posible utilizar un selector más avanzado:**

```
.error { color: red; }
.error.destacado { color: blue; }
.destacado { font-size: 15px; }
.especial { font-weight: bold; }
<p class="especial destacado error">Párrafo de texto...</p>
```

En el ejemplo anterior, el color de la letra del texto es azul y no rojo. El motivo es que se ha utilizado un selector de clase múltiple .error.destacado, que se interpreta como "aquellos elementos de la página que dispongan de un atributo class con al menos los valores error y destacado".

2.2. 4 Selector identificador

El **selector identificador** utiliza el atributo id para seleccionar un elemento. Solo puede haber un elemento con un id dado en un documento.

Sintaxis: #id { atributo:valor }

Ejemplo:

```
#destacado { color: red; }

<p>Primer párrafo</p>
<p id="destacado">Segundo párrafo</p>
<p>Tercer párrafo</p>
```

En el ejemplo anterior, el selector #destacado solamente selecciona el segundo párrafo (cuyo atributo id es igual a destacado).

La principal **diferencia entre este tipo de selector y el selector de clase** tiene que ver con HTML y no con CSS. Como se sabe, **en una misma página, el valor del atributo id debe ser único, de forma que dos elementos diferentes no pueden tener el mismo valor de id**. Sin embargo, **el atributo class no es obligatorio que sea único**, de forma que muchos elementos HTML diferentes pueden compartir el mismo valor para su atributo class.

De esta forma, la **recomendación** general es la de **utilizar el selector de ID cuando se quiere aplicar un estilo a un solo elemento específico de la página y utilizar el selector de clase cuando se quiere aplicar un estilo a varios elementos diferentes** de la página HTML.

Al igual que los selectores de clase, en este caso también se puede **restringir el alcance del selector** mediante la combinación con otros selectores. El siguiente ejemplo aplica la regla CSS solamente al elemento de tipo <p> que tenga un atributo id igual al indicado:

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

A primera vista, restringir el alcance de un selector de ID puede parecer absurdo. En realidad, un selector de tipo p#aviso sólo tiene sentido cuando el archivo CSS se aplica sobre muchas páginas HTML diferentes.

En este caso, algunas páginas pueden disponer de elementos con un atributo id igual a aviso y que no sean párrafos, por lo que la regla anterior no se aplica sobre esos elementos.

No debe confundirse el selector de ID con los selectores anteriores:

```
/* Todos los elementos de tipo "p" con atributo id="aviso" */
p#aviso { ... }
/* Todos los elementos con atributo id="aviso" que estén dentro de cualquier elemento de tipo "p" */
p #aviso { ... }
/* Todos los elementos "p" de la página y todos los elementos con atributo id="aviso" de la página */
p, #aviso { ... }
```

2.2.5 Selector descendiente

Un **elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento padre**.

Sintaxis: selector1 selector2 selectorN {atributo: valor;} /* El estilo se aplica sobre el selector N */

Ejemplo: div p { color: black;} /* El estilo se aplica a todos los párrafos que se encuentren dentro de una etiqueta div */

Ejemplo: p span { color: red; }

Si el código HTML de la página es el siguiente:

```
<p>
...
<span>texto1</span>
...
<a href="">...<span>texto2</span></a>
...
```

</p>

El selector `p span` selecciona tanto `texto1` como `texto2`. El motivo es que, **en el selector descendente, un elemento no tiene que ser descendiente directo del otro. La única condición es que un elemento debe estar dentro de otro elemento, sin importar el nivel de profundidad en el que se encuentre.** Al resto de elementos `` de la página que no están dentro de un elemento `<p>`, no se les aplica la regla CSS anterior.

Ejemplo: el selector descendente se compone de cuatro selectores:

```
p a span em { text-decoration: underline; }
```

Los estilos de la regla anterior **se aplican a los elementos de tipo `` que se encuentren dentro de elementos de tipo ``, que a su vez se encuentren dentro de elementos de tipo `<a>` que se encuentren dentro de elementos de tipo `<p>`.**

No debe confundirse el selector descendente con la combinación de selectores:

2.2.6 Combinación de selectores

La combinación de selectores nos permite dar un **estilo a todos los selectores indicados.**

Sintaxis: `selector1, selector2, selector3{atributo: valor;}` /* El estilo se aplica sobre los selectores indicados */

Ejemplo: el selector combinado se compone de cuatro selectores:

```
p, a, span, em { text-decoration: underline; } /* El estilo se aplica a todos los elementos p, a,span y em */
```

Se puede restringir el alcance del selector descendente combinándolo con el selector universal.

Ejemplo: se muestran los dos enlaces de color rojo:

```
p a { color: red; }
<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>
```

Sin embargo, en el siguiente ejemplo **solamente el segundo enlace se muestra de color rojo:**

```
p * a { color: red; }
<p><a href="#">Enlace</a></p>
<p><span><a href="#">Enlace</a></span></p>
```

La razón es que el selector `p * a` se interpreta como **todos los elementos de tipo `<a>` que se encuentren dentro de cualquier elemento que, a su vez, se encuentre dentro de un elemento de tipo `<p>`.** Como el primer elemento `<a>` se encuentra directamente bajo un elemento `<p>`, no se cumple la condición del selector `p * a`.

2.2.7 Selector de hijos

Se trata de un selector similar al selector descendente, pero muy diferente en su funcionamiento. Se usa para **seleccionar un elemento que es *hijo directo* de otro elemento y se indica con el signo `>`.**

Sintaxis: selector1 > selector2 {atributo: valor;} /* El estilo se aplica sobre el selector 2 */

Ejemplo:

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

<p><span>Texto1</span></p>
<p><a href="#"><span>Texto2</span></a></p>
```

En el ejemplo anterior, el selector `p > span` se interpreta como **"cualquier elemento `` que sea hijo directo de un elemento `<p>`"**, por lo que el primer elemento `` cumple la condición del selector. Sin embargo, el segundo elemento `` no la cumple porque es descendiente pero no es hijo directo de un elemento `<p>`.

Ejemplo: muestra las diferencias entre el selector descendente y el selector de hijos:

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

<p><a href="#">Enlace1</a></p>
<p><span><a href="#">Enlace2</a></span></p>
```

El **primer selector es de tipo descendente** y por tanto se aplica a todos los elementos `<a>` que se encuentran dentro de elementos `<p>`. En este caso, los estilos de este selector se aplican a los dos enlaces.

Por otra parte, **el selector de hijos obliga a que el elemento `<a>` sea hijo directo de un elemento `<p>`**. Por lo tanto, los estilos del selector `p > a` no se aplican al segundo enlace del ejemplo anterior.

2.2.8 Selector adyacente

Se usa para **seleccionar elementos que son *hermanos*, es decir, su elemento padre es el mismo y están seguidos en el código HTML**.

Sintaxis: selector1 + selector2 { atributo: valor; } /* El estilo se aplica al selector 2 */

Ejemplo: Si se considera el siguiente código HTML:

```
<body>
  <h1>Titulo1</h1>
  <h2>Subtítulo</h2>
  ...
  <h2>Otro subtítulo</h2>
  ...
</body>
```

La página anterior dispone de dos elementos `<h2>`, pero sólo uno de ellos se encuentra inmediatamente después del elemento `<h1>`. Si se quiere aplicar diferentes colores en función de esta circunstancia, el selector adyacente es el más adecuado:

```
h2 { color: green; }
h1 + h2 { color: red; }
```

Las reglas CSS anteriores hacen que todos los <h2> de la página se vean de color verde, salvo aquellos <h2> que se encuentran inmediatamente después de cualquier elemento <h1> y que se muestran de color rojo.

2.2.9 Selector de atributos

El último tipo de selectores avanzados lo forman los selectores de atributos, que permiten seleccionar elementos HTML en función de sus atributos y/o valores de esos atributos.

Los cuatro tipos de selectores de atributos son:

- **[nombre_atributo]**, selecciona los elementos que tienen establecido el atributo llamado nombre_atributo, independientemente de su valor.
- **[nombre_atributo=valor]**, selecciona los elementos que tienen establecido un atributo llamado nombre_atributo con un valor igual a valor.
- **[nombre_atributo^=valor]**, selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y su valor comience por valor
- **[nombre_atributo\$=valor]**, selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y su valor finaliza por valor
- **[nombre_atributo~=valor]**, selecciona los elementos que tienen establecido un atributo llamado nombre_atributo y al menos uno de los valores del atributo es valor.

Ejemplos de estos tipos de selectores:

```
/* Se muestran de color azul todos los enlaces que tengan un atributo "class", independientemente de su valor */
a[class] { color: blue; }
/* Se muestran de color azul todos los enlaces que tengan un atributo "class" con el valor "externo" */
a[class="externo"] { color: blue; }
/* Se muestran de color azul todos los enlaces que apunten al sitio "http://www.ejemplo.com" */
a[href="http://www.ejemplo.com"] { color: blue; }
/* Se muestran de color azul todos los enlaces que tengan un atributo "class" en el que al menos uno de sus valores sea "externo" */
a[class~="externo"] { color: blue; }
/* Selecciona todos los elementos de la página cuyo atributo "lang" sea igual a "en", es decir, todos los elementos en inglés */
*[lang=en] { ... }
```

Resumen de selectores básicos

Selector	Descripción
*	Selecciona todos los elementos del DOM
etiqueta	Selecciona todas las etiquetas indicadas

.class	Selección de los elementos con la clase .class
#id	Selección del elemento con id #id
sel1 sel2	Selección de los selectores sel2 que se encuentren dentro de los selectores sel1
.class1.class2	Selección de los elementos con las dos clases: class1 y class2
sel1.class1	Selección de todos los selectores sel1 con clase class1
sel1, sel2	Selección de todos los selectores separados por comas
sel1 > sel2	Selección de los selectores sel2 cuando son hijos de sel1
sel1 + sel2	Selección del selector sel2 cuando es hermano de sel1 (su elemento padre es el mismo)

2.2.10 Ejemplos

styles.css

```
p { background-color: grey; } /* Selector etiqueta */
.clase { color: red; } /* Selector clase */
#ident { color: green; } /* Selector Identificador */
* { font-style: italic; } /* Selector universal */
p a { background-color: orange; } /* Selector descendiente */
h3, small { color: blue; } /* Combinación de selectores */
div>span { color: pink; } /* Selector de hijos */
span+small { background-color: yellow; } /* Selector adyacente*/
```

inicio.html

```
<html>
<head>
  <meta charset="utf-8">
  <title>Selectores</title>
  <link rel="stylesheet" href="styles.css">
</head>
<body>
  <h1 class="clase">Texto de color rojo</h1>
  <p id="ident">Texto de color verde</p>
  <h2>Selector descendiente</h2>
  <p>
    <a href="#">Enlace</a><br>
    <span>
      <a href="#">Enlace</a>
    </span>
  </p>
  <h3>Texto h3</h3>
  <div>
    <span>Texto con span</span>
    <small> Texto con small</small>
    <span>Texto con span</span>
    <small> Texto con small</small>
    <small> Texto con small</small>
  </div>
</body>
</html>
```



2.2.11 AGRUPACIÓN DE REGLAS.

Cuando se crean archivos CSS complejos con decenas o cientos de reglas, es habitual que los estilos que se aplican a un mismo selector se definan en diferentes reglas:

```
h1 { color: red; }  
...  
h1 { font-size: 2em; }  
...  
h1 { font-family: Verdana; }
```

Las tres reglas anteriores establecen el valor de tres propiedades diferentes de los elementos `<h1>`. Antes de que el navegador muestre la página, procesa todas las reglas CSS de la página para tener en cuenta todos los estilos definidos para cada elemento.

Cuando el selector de dos o más reglas CSS es idéntico, se pueden **agrupar las declaraciones de las reglas para hacer las hojas de estilos más eficientes**:

```
h1 {  
  color: red;  
  font-size: 2em;  
  font-family: Verdana;  
}
```

El ejemplo anterior tiene el mismo efecto que las tres reglas anteriores, pero es más eficiente y es más fácil de modificar y mantener por parte de los diseñadores. **Como CSS ignora los espacios en blanco y las nuevas líneas, también se pueden agrupar las reglas de la siguiente forma:**

```
h1 { color: red; font-size: 2em; font-family: Verdana; }
```

Si se quiere **reducir al máximo el tamaño del archivo CSS** para mejorar ligeramente el tiempo de carga de la página web, también es posible indicar la regla anterior de la siguiente forma:

```
h1 {color:red;font-size:2em;font-family:Verdana;}
```

2.2.12 HERENCIA.

Una de las características principales de CSS es la **herencia de los estilos definidos para los elementos**. Cuando se establece el valor de una propiedad CSS en un elemento, sus elementos descendientes heredan de forma automática el valor de esa propiedad. Si se considera el siguiente ejemplo:

```
<!DOCTYPE html >  
<head>  
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />  
  <title>Ejemplo de herencia de estilos</title>  
  <style type="text/css">  
    body { color: blue; }  
  </style>  
</head>  
  
<body>
```

```
<h1>Titular de la página</h1>
<p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

En el ejemplo anterior, el selector **body** solamente establece el color de la letra para el elemento **<body>**. No obstante, **la propiedad color es una de las que se heredan de forma automática, por lo que todos los elementos descendientes de <body> muestran ese mismo color de letra.** Por tanto, establecer el color de la letra en el elemento **<body>** de la página implica cambiar el color de letra de todos los elementos de la página.

Aunque **la herencia de estilos se aplica automáticamente, se puede anular su efecto estableciendo de forma explícita otro valor para la propiedad que se hereda**, como se muestra en el siguiente ejemplo:

```
<!DOCTYPE html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Ejemplo de herencia de estilos</title>
<style type="text/css">
  body { font-family: Arial; color: black; }
  h1 { font-family: Verdana; }
  p { color: red; }
</style>
</head>

<body>
  <h1>Titular de la página</h1>
  <p>Un párrafo de texto no muy largo.</p>
</body>
</html>
```

En el ejemplo anterior, **se establece en primer lugar el color y tipo de letra del elemento <body>, por lo que todos los elementos de la página se mostrarían con ese mismo color y tipo de letra. No obstante, las otras reglas CSS modifican alguno de los estilos heredados.**

De esta forma, los elementos **<h1>** de la página se muestran con el tipo de letra Verdana establecido por el selector **h1** y se muestran de color negro que es el valor heredado del elemento **<body>**. Igualmente, los elementos **<p>** de la página se muestran del color rojo establecido por el selector **p** y con un tipo de letra Arial heredado del elemento **<body>**.

La mayoría de propiedades CSS aplican la herencia de estilos de forma automática. Además, para aquellas propiedades que no se heredan automáticamente, CSS incluye un mecanismo para forzar a que se hereden sus valores, tal y como se verá más adelante.

Por último, aunque la herencia automática de estilos puede parecer complicada, **simplifica en gran medida la creación de hojas de estilos complejas.** Como se ha visto en los ejemplos anteriores,

si se quiere establecer por ejemplo la tipografía base de la página, simplemente se debe establecer en el elemento `<body>` de la página y el resto de elementos la heredarán de forma automática.

2.2.12 COLISIONES DE ESTILOS

En las hojas de estilos complejas, es habitual **que varias reglas CSS se apliquen a un mismo elemento HTML**. El problema de estas reglas múltiples es que se pueden dar colisiones como la del siguiente ejemplo:

```
p { color: red; }  
p { color: blue; }  
  
<p>...</p>
```

¿De qué color se muestra el párrafo anterior? **CSS tiene un mecanismo de resolución de colisiones muy complejo y que tiene en cuenta el tipo de hoja de estilo que se trate (de navegador, de usuario o de diseñador), la importancia de cada regla y lo específico que sea el selector.**

El **método** seguido por CSS para **resolver las colisiones de estilos** se muestra a continuación:

1. Determinar todas las declaraciones que se aplican al elemento para el medio CSS seleccionado.
2. **Ordenar** las declaraciones **según su origen** (CSS de navegador, de usuario o de diseñador) y su prioridad (palabra clave `!important`).
3. **Ordenar** las declaraciones **según lo específico que sea el selector**. Cuanto más genérico es un selector, menos importancia tienen sus declaraciones.
4. Si después de aplicar las normas anteriores existen dos o más reglas con la misma prioridad, **se aplica la que se indicó en último lugar**.

El mecanismo simplificado que se puede aplicar es el siguiente:

1. Cuanto más específico sea un selector, más importancia tiene su regla asociada.
2. A igual **especificidad**, se considera la última regla indicada.

Como en el ejemplo anterior los dos selectores son idénticos, las dos reglas tienen la misma prioridad y prevalece la que se indicó en último lugar, por lo que el párrafo se muestra de color azul.

En el siguiente ejemplo, la regla CSS que prevalece se decide por lo específico que es cada selector:

```
p { color: red; }  
p#especial { color: green; }  
* { color: blue; }  
<p id="especial">...</p>
```

Al elemento <p> se le aplican las tres declaraciones. **Como su origen y su importancia es la misma, decide la especificidad del selector.**

- El selector * es el menos específico, ya que se refiere a "todos los elementos de la página".
- El selector p es poco específico porque se refiere a "todos los párrafos de la página".
- Por último, el selector p#especial sólo hace referencia a "el párrafo de la página cuyo atributo id sea igual a especial".

Como el selector **p#especial es el más específico, su declaración es la que se tiene en cuenta y por tanto el párrafo se muestra de color verde.**