

# Selectores CSS

## Selectores Básicos

### 2.1.1. Selector universal

Se utiliza para seleccionar todos los elementos de la página. El siguiente ejemplo elimina el margen y el relleno de todos los elementos HTML (por ahora no es importante fijarse en la parte de la declaración de la regla CSS):

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

El selector universal se indica mediante un asterisco (\*). A pesar de su sencillez, no se utiliza habitualmente, ya que es difícil que un mismo estilo se pueda aplicar a todos los elementos de una página.

No obstante, sí que se suele combinar con otros selectores y además, forma parte de algunos *hacks* muy utilizados, como se verá más adelante.

### 2.1.2. Selector de tipo o etiqueta

Selecciona todos los elementos de la página cuya etiqueta HTML coincide con el valor del selector. El siguiente ejemplo selecciona todos los párrafos de la página:

```
p {  
  
    ...  
  
}
```

Para utilizar este selector, solamente es necesario indicar el nombre de una etiqueta HTML (sin los caracteres < y >) correspondiente a los elementos que se quieren seleccionar.

El siguiente ejemplo aplica diferentes estilos a los titulares y a los párrafos de una página HTML:

```
h1 {  
  
    color: red;  
  
}
```

```
h2 {  
  
    color: blue;  
  
}
```

```
p {  
  
    color: black;  
  
}
```

Si se quiere aplicar los mismos estilos a dos etiquetas diferentes, se pueden encadenar los selectores. En el siguiente ejemplo, los títulos de sección `h1`, `h2` y `h3` comparten los mismos estilos:

```
h1 {  
  
    color: #8A8E27;  
  
    font-weight: normal;  
  
    font-family: Arial, Helvetica, sans-serif;  
  
}
```

```
h2 {  
  
    color: #8A8E27;  
  
    font-weight: normal;  
  
    font-family: Arial, Helvetica, sans-serif;  
  
}
```

```
h3 {
```

```
color: #8A8E27;

font-weight: normal;

font-family: Arial, Helvetica, sans-serif;

}
```

En este caso, CSS permite agrupar todas las reglas individuales en una sola regla con un selector múltiple. Para ello, se incluyen todos los selectores separados por una coma (,) y el resultado es que la siguiente regla CSS es equivalente a las tres reglas anteriores:

```
h1, h2, h3 {

    color: #8A8E27;

    font-weight: normal;

    font-family: Arial, Helvetica, sans-serif;

}
```

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-weight: normal;

    font-family: Arial, Helvetica, sans-serif;

}
```

```
h1 { font-size: 2em; }
```

```
h2 { font-size: 1.5em; }
```

```
h3 { font-size: 1.2em; }
```

### 2.1.3. Selector descendente

Selecciona los elementos que se encuentran dentro de otros elementos. Un elemento es descendiente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del otro elemento.

El selector del siguiente ejemplo selecciona todos los elementos `<span>` de la página que se encuentren dentro de un elemento `<p>`:

```
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 `<span>` de la página que no están dentro de un elemento `<p>`, no se les aplica la regla CSS anterior.

Los selectores descendentes permiten aumentar la precisión del selector de tipo o etiqueta. Así, utilizando el selector descendente es posible aplicar diferentes estilos a los elementos del mismo tipo. El siguiente ejemplo amplía el anterior y muestra de color azul todo el texto de los `<span>` contenidos dentro de un `<h1>`:

```
p span { color: red; }
```

```
h1 span { color: blue; }
```

Con las reglas CSS anteriores:

- Los elementos `<span>` que se encuentran dentro de un elemento `<p>` se muestran de color rojo.
- Los elementos `<span>` que se encuentran dentro de un elemento `<h1>` se muestran de color azul.
- El resto de elementos `<span>` de la página, se muestran con el color por defecto aplicado por el navegador.

La sintaxis formal del selector descendente se muestra a continuación:

```
selector1 selector2 selector3 ... selectorN
```

Los selectores descendentes siempre están formados por dos o más selectores separados entre sí por espacios en blanco. El último selector indica el elemento sobre el que se aplican los estilos y todos los selectores anteriores indican el lugar en el que se debe encontrar ese elemento.

En el siguiente 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 `<em>` que se encuentren dentro de elementos de tipo `<span>`, 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:

```
/* El estilo se aplica a todos los elementos "p", "a", "span" y "em" */
```

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

```
/* El estilo se aplica solo a los elementos "em" que se
```

```
encuentran dentro de "p a span" */
```

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

Se puede restringir el alcance del selector descendente combinándolo con el selector universal. El siguiente ejemplo, muestra 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`.

#### 2.1.4. Selector de clase

Si se considera el siguiente código HTML de ejemplo:

```
<body>
```

```
<p>Lorem ipsum dolor sit amet...</p>
```

```
<p>Nunc sed lacus et est adipiscing accumsan...</p>
```

```
<p>Class aptent taciti sociosqu ad litora...</p>
```

```
</body>
```

A continuación se muestra otro ejemplo de selectores de clase:

```
.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 `<span>` 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`"*. De la misma forma, el selector `a.destacado` solamente selecciona los enlaces cuyo atributo `class` sea igual a `destacado`.

De lo anterior se deduce que el atributo `.destacado` es equivalente a `*.destacado`, por lo que todos los diseñadores obvian el símbolo `*` al escribir un selector de clase normal.

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. En el siguiente ejemplo:

```
<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 siguiente ejemplo, el texto del párrafo se vería de color rojo, en negrita y con un tamaño de letra de 15 píxel:

```
.error { color: red; }
```

```
.destacado { font-size: 15px; }
```

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

```
<p class="especial destacado error">Párrafo de texto...</p>
```

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`"*.

### 2.1.5. Selectores de ID

En ocasiones, es necesario aplicar estilos CSS a un único elemento de la página. Aunque puede utilizarse un selector de clase para aplicar estilos a un único elemento, existe otro selector más eficiente en este caso.

El selector de ID permite seleccionar un elemento de la página a través del valor de su atributo `id`. Este tipo de selectores sólo seleccionan un elemento de la página porque el valor del atributo `id` no se puede repetir en dos elementos diferentes de una misma página.

La sintaxis de los selectores de ID es muy parecida a la de los selectores de clase, salvo que se utiliza el símbolo de la almohadilla (#) en vez del punto (.) como prefijo del nombre de la regla CSS:

```
#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 { ... }
```

### 2.1.6. Combinación de selectores básicos

CSS permite la combinación de uno o más tipos de selectores para restringir el alcance de las reglas CSS. A continuación se muestran algunos ejemplos habituales de combinación de selectores.

```
.aviso .especial { ... }
```

El anterior selector solamente selecciona aquellos elementos con un `class="especial"` que se encuentren dentro de cualquier elemento con un `class="aviso"`.

Si se modifica el anterior selector:

```
div.aviso span.especial { ... }
```

Ahora, el selector solamente selecciona aquellos elementos de tipo `<span>` con un atributo `class="especial"` que estén dentro de cualquier elemento de tipo `<div>` que tenga un atributo `class="aviso"`.

La combinación de selectores puede llegar a ser todo lo compleja que sea necesario:

```
ul#menuPrincipal li.destacado a#inicio { ... }
```

## 2.2. Selectores avanzados

Utilizando solamente los selectores básicos de la sección anterior, es posible diseñar prácticamente cualquier página web. No obstante, CSS define otros selectores más avanzados que permiten simplificar las hojas de estilos.

### 2.2.1. Selector de hijos

Se trata de un selector similar al selector descendente, pero muy diferente en su funcionamiento. Se utiliza para seleccionar un elemento que es *hijo directo* de otro elemento y se indica mediante el "*signo de mayor que*" (`>`):

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

```
<p><span>Texto1</span></p>
```

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

### 2.2.2. Selector adyacente

El selector adyacente se emplea para seleccionar elementos que en el código HTML de la página se encuentran justo a continuación de otros elementos. Su sintaxis emplea el signo `+` para separar los dos elementos:

```
elemento1 + elemento2 { ... }
```

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 }
```

### 2.2.3. 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 al menos uno de los valores del atributo es `valor`.
- `[nombre_atributo|=valor]`, selecciona los elementos que tienen establecido un atributo llamado `nombre_atributo` y cuyo valor es una serie de palabras separadas con guiones, pero que comienza con `valor`. Este tipo de selector sólo es útil para los atributos de tipo `lang` que indican el idioma del contenido del elemento.

A continuación se muestran algunos ejemplos de estos tipos de selectores:

```
/* Se muestran d 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] { ... }

/* Selecciona todos los elementos de la página cuyo atributo

    "lang" empiece por "es", es decir, "es", "es-ES", "es-AR", etc. */

*[lang|="es"] { color : red }

```

## Propiedades CSS

### Propiedades de fuentes

- Font Family (Familia de fuente)
- Font Style (Estilo de fuente)
- Font Variant (Variante de fuente)

- Font Weight (Peso de fuente)
- Font Size (Tamaño de fuente)
- Font (Fuente)

#### Propiedades de color y fondo

- Color
- Background Color (Color de fondo)
- Background Image (Imagen de fondo)
- Background Repeat (Repetir fondo)
- Background Attachment (Fijación de fondo)
- Background Position (Posición del fondo)
- Background (Fondo)

#### Propiedades de texto

- Word Spacing (Espaciamiento de palabras)
- Letter Spacing (Espaciamiento de letras)
- Text Decoration (Decoración de texto)
- Vertical Alignment (Alineación vertical)
- Text Transformation (Transformación de texto)
- Text Alignment (Alineación de texto)
- Text Indentation (Sangría de texto)
- Line Height (Altura de línea)

#### Propiedades de cuadros

- Top Margin (Margen superior)
- Right Margin (Margen derecho)
- Bottom Margin (Margen inferior)
- Left Margin (Margen izquierdo)
- Margin (Margen)
- Top Padding (Relleno superior)
- Right Padding (Relleno derecho)
- Bottom Padding (Relleno inferior)
- Left Padding (Relleno izquierdo)
- Padding (Relleno)
- Top Border Width (Ancho de borde superior)
- Right Border Width (Ancho de borde derecho)
- Bottom Border Width (Ancho de borde inferior)
- Left Border Width (Ancho de borde izquierdo)
- Border Width (Ancho de borde)
- Border Color (Color de borde)
- Border Style (Estilo de borde)
- Top Border (Borde superior)
- Right Border (Borde derecho)
- Bottom Border (Borde de fondo)
- Left Border (Borde izquierdo)
- Border (Borde)
- Width (Ancho)
- Height (Altura)

- [Float \(Flotar\)](#)
- [Clear \(Despejar\)](#)

#### Propiedades de clasificación

- [Display \(Visualización\)](#)
  - [Whitespace \(Espacio en blanco\)](#)
  - [List Style Type \(Tipo de estilo de lista\)](#)
  - [List Style Image \(Imagen de estilo de lista\)](#)
  - [List Style Position \(Posición de estilo de lista\)](#)
  - [List Style \(Estilo de lista\)](#)
- 

## Agrupaciones HTML

Hasta ahora hemos visto cómo insertar diferentes elementos sobre un documento [HTML](#). Estos elementos se irán mostrando según la secuencia en la que hayamos escrito el documento [HTML](#).

Una de las cosas que tenemos que saber de los elementos html es si son elementos de bloque o elementos de línea.

Un **elemento de bloque** es aquél que una vez utilizado aparece en la siguiente línea y ocupa todo el ancho.

Elementos de tipo bloque son los [párrafos p](#), los [formularios form](#), o [las cabeceras hx](#).

Un **elemento en línea** es aquel que se muestra justo a continuación del anterior elemento. Estos elementos serían los [enlaces a](#), [imágenes img](#),...

El lenguaje [HTML](#) nos permite agrupar un conjunto de elementos mediante una agrupación en bloque o una agrupación en línea.

### Agrupaciones en Bloque

Un elemento en bloque siempre empieza con una línea y su tamaño será igual al ancho disponible. El ancho disponible inicialmente es el de la página.

El elemento que nos permite realizar agrupaciones en bloque es el [elemento div](#) o más conocidos como capas. La estructura del [elemento div](#) es:

```
<div>
<!-- Contenido de La Capa -->
</div>
```

Los elementos en bloque pueden contener a otros elementos en bloque o bien a otros elementos en línea.

Por ejemplo podríamos agrupar en un bloque el siguiente contenido.

```
<div id="micapa">
  <h2>Título del Contenido</h2>
  Este es el contenido del artículo
  
```

```
<p>Más contenido del artículo</p>
</div>
```

### Agrupaciones en Línea

Para poder realizar agrupaciones en línea tenemos el [elemento span](#). La estructura del [elemento span](#) será:

```
<span> <!-- Contenido --></span>
```

Las agrupaciones en línea sólo pueden contener a otros elementos en línea, no a elementos de tipo bloque.

Por ejemplo podríamos tener la siguiente agrupación en línea.

```
<span id="entrada">
  <strong>Artículo Nuevo</strong>,
  <em>,12 de marzo de 2016</em>
</span>
```

Es muy normal que los agrupadores, ya sean o bien [div](#), o bien [span](#) lleven el [atributo id](#) o [class](#), ya que a posteriori serán manipulados mediante [hojas de estilo CSS](#) utilizando dichos identificadores.