

Curso Web accesible con XHTML y CSS. Bloque III. Hojas de Estilo en Cascada: Selectores.

Cursos Thales CICA Web 2006.

Curso WEBCSS

A. Gámez, L.M. Marín, F. Mesa & S. Fandiño

Índice del Bloque III. Hojas de Estilo en Cascada: Selectores.

Selectores

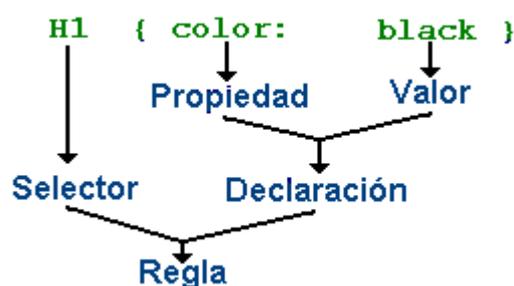
Índice

1. Definición de reglas de estilo
 2. Sintaxis general de las reglas
 3. Selectores. Elementos que se pueden seleccionar con CSS
 4. Tipos de selectores
 5. Selectores básicos
 6. Selectores Genealógicos
 7. Pseudo-Selectores
 8. Selectores de atributos
-

1. Definición de reglas de estilo

Como vimos en el tema anterior, una hoja de estilo se compone de reglas de visualización (reglas de estilo). Aunque se puede indicar un estilo directamente a un elemento de la página mediante la definición de estilos singulares dentro de la página HTML, que definen la apariencia de una sola etiqueta HTML (por ejemplo un párrafo, dentro de la página) nos centraremos en la definición de estilos mediante la llamada a un archivo externo (bien usando `link` o usando `@import`) o en la cabecera del documento (mediante la etiqueta `<style>`). El método aconsejable es mediante links a un archivo `.css` externo. Este es el sistema mas ortodoxo, según la filosofía de las CSS, pues separa totalmente lo que es contenido (en el documento HTML) de lo que es forma (en el archivo CSS).

Cada regla consta de un **SELECTOR**, que es el que indica a qué elemento o parte de la página se aplica el estilo; normalmente los selectores son etiquetas HTML (p.e. `h1` sería un selector que permitiría aplicar estilos a las cabeceras `<h1>`). A cada selector debe seguir una **DECLARACIÓN** del estilo que ha de serle aplicado (p.e. `color:blue`). Toda declaración tiene dos partes: **PROPIEDAD** (por ejemplo, `color`) y **VALOR** (por ejemplo, `blue`).



Los estilos, entonces, se definen así:

```

selector { propiedad:valor }
h1 {color : blue}
  
```

Por tanto, para poder declarar reglas, deberemos conocer lo siguiente:

- La sintaxis general de las reglas.
- Qué valores podemos indicar en **SELECTOR**.
- Qué valores podemos indicar en **PROPIEDAD** para cada **SELECTOR**.
- Qué valores podemos indicar en **VALOR** para cada **PROPIEDAD**.

Los **selectores** disponibles son, inicialmente, **todas** las etiquetas HTML. A cualquier etiqueta HTML se le puede asignar un estilo; además como veremos en otra sección, se pueden crear selectores propios. Las **propiedades** disponibles son las enunciadas como tales en el standard CSS; y los **valores** son asimismo los asignados en dicho standard. O dicho de

otro modo: solo pueden utilizarse propiedades y valores contemplados en el standard, no siendo posible crear propiedades y valores.

2. Sintaxis general de las reglas

El enunciado de reglas constituye la forma usual de declarar estilos, y deben ajustarse a la sintaxis definida por la especificación CSS; si el navegador encuentra un selector que no entiende, ignorará la declaración por completo. Si encuentra dentro de la declaración una propiedad o valor que no comprende ignorará esta parte de la declaración, pero deberá procesar el resto.

Los **selectores** se escriben omitiendo las llaves < y >, es decir, simplemente h1, h2, etc. La **declaración** {propiedad : valor} ha de ir encerrada en llaves { }.

Pueden asignarse varias propiedades a un mismo selector, separadas por punto y coma:

```
selector { propiedad1: valor1; propiedad2: valor2 }
h1 {color: yellow; background-color: red }
```

También se pueden asignar propiedades de forma conjunta a varios selectores, **separándolos por comas**:

```
selector1, selector2 {propiedad1:valor1;propiedad2:valc
```

Como ejemplo, veamos una definición de un estilo para los elementos (selectores) h1 y h2:

```
<head>
  <title>Titulo</title>
  <style>
    h1 { font-size: x-large; color: red; }
    h2 { font-size: large; color: blue; }
  </style>
</head>
```

Se pueden añadir líneas y espacios en blanco entre declaraciones para mejorar la legibilidad. También se pueden añadir comentarios. En las hojas de estilo externas el único formato de comentario admisible es /* aquí dentro va el comentario */

3. Selectores. Elementos que se pueden seleccionar con CSS

Los selectores más simples se aplican a etiquetas html. Sin embargo, se pueden usar selectores más sofisticados. Como selectores pueden usarse los siguientes elementos:

- * Todos los elementos de un tipo determinado: por ejemplo todos los encabezados de primer nivel (h1).
- * Todos los elementos de un tipo determinado que pertenezcan a la misma clase (luego veremos como asignar clases a los elementos)
- * Todos los elementos de determinado tipo que estén dentro de otros elementos que nosotros especifiquemos. (por ejemplo, todos los enlaces que formen parte de una lista).
- * Todos los elementos de cierto tipo que estén dentro de otro y que además pertenezcan a una clase. Por ejemplo, todos los enlaces que estén dentro de una lista y que además pertenezcan a la clase "pepe".
- * Todos los elementos de cierto tipo, pero sólo cuando aparezcan después de un determinado elemento. Por ejemplo todos los párrafos que aparezcan después de un elemento h2.
- * Cualquier elemento que pertenezca a determinada clase.
- * Un elemento específico al que tenga asignado un determinado ID.

En las siguientes secciones veremos los distintos casos.

4. Tipos de selectores

Desde un punto de vista lógico, podemos usar la siguiente clasificación de los selectores:

Selectores básicos:

Selector universal

Selectores de etiqueta

Selectores de clases

Selectores de ID

Selectores agrupados

Selectores Genealógicos:

Selectores de elementos descendientes

Selectores de hijos

Selectores de primogénitos

Selectores de hermanos adyacentes

Pseudo-selectores:

Pseudo-clases de enlaces

Pseudo-clases dinámicas

Pseudo-clases de idioma

Pseudo-elementos

Selectores de atributos

Pasemos a ver los distintos selectores atendiendo a esta clasificación.

5. Selectores básicos

Selector universal

Este selector se utiliza para asignar propiedades a todos los elementos de una página. Para ello se utiliza un asterisco (*). Ejemplo

```
* { color: red }
```

Selectores de etiqueta

Los selectores de etiqueta seleccionan cualquier elemento de una página que coincida con el selector. Por ejemplo, el selector p se aplicaría a todos los párrafos que aparecieran en una página. Para especificar un selector de etiqueta, simplemente utilizamos la etiqueta sin los <> Ejemplo:

```
p {font-size: small;
```

```
text-align: justify;}
```

A partir de la utilización del estilo anterior, todos los párrafos tendrán tipo de letra pequeña y estarán justificados. En el siguiente ejemplo:

```
body { color: red}  
p {font-size: small;  
text-align: justify;}
```

El color de letra en los párrafos sería rojo, aunque no se indica, ya que hereda esta propiedad de su elemento padre **body**.

Selectores de clases

Cuando se declara una regla en la hoja de estilo, esta regla se aplica a todos los elementos HTML descritos por su selector. Así en los ejemplos anteriores se están indicando las propiedades de todos los párrafos del documento. Sin embargo, en muchas ocasiones se pretende destacar algunos párrafos y darles características distintas a la del resto. Esto podría conseguirse añadiendo etiquetas html del tipo ``, pero esto es precisamente lo que estamos tratando de evitar. Para ello, se provee de un atributo extra en casi todos los elementos del lenguaje HTML, llamado **class** (clase). Podemos organizar "clases" de elementos del mismo tipo y aplicar una apariencia distinta a cada clase.

Supongamos que tenemos una página formada por preguntas y respuestas. Tendríamos, por tanto, dos tipos distintos de párrafos. Podríamos poner las preguntas en rojo y las respuestas en azul, mientras que otros párrafos tendrían otras propiedades. Utilizando las clases podemos asignar a los párrafos que sean una pregunta, la clase pregunta, y a los que sean una respuesta la clase respuesta. Y luego, en la hoja de estilos asignar una apariencia distinta a cada una.

El código html sería

```
<p class="pregunta"> ¿Qué extensión tienen las hojas  
de estilo? </p>  
<p class="respuesta"> css </p>
```

Mientras que en la hoja de estilo tendríamos:

```
p.pregunta {color: red;}  
p.respuesta {color:blue;}
```

En este ejemplo estamos indicando que todos los párrafos que sean de la clase pregunta serán en rojo y los de clase respuesta serán en azul. En este ejemplo si tuviéramos una cabecera h3 de la clase pregunta

```
<h3 class="pregunta"> ¿Qué extensión tienen las hojas </h3>
```

no se le aplicaría el color rojo, ya que en la hoja de estilo se está indicando p.pregunta, es decir, que la orden solo afecta la clase pregunta de la etiqueta p, no de la etiqueta h3.

En cambio, si en la hoja de estilo pusiéramos

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

estaríamos aplicando el color rojo a todos los elementos de la clase pregunta, independientemente de en qué etiqueta estén.

Un error muy frecuente es dar un nombre a la clase relativo a la apariencia visual, en lugar de a su función. En el ejemplo anterior se está dando nombre según la función: "preguntas" o "respuestas". Sin embargo se podría haber dado como nombre a las clases "rojo" y "azul". Evidentemente, podemos elegir el nombre que queramos, pero si en el futuro decidimos cambiar y que las preguntas en vez de ser en rojo sean verdes, tendremos el lío de que la clase "rojo" no sale de color rojo, sino verde.

Selectores de id

Los selectores de id funcionan de manera muy similar a los selectores de clases, salvo que, a diferencia de estos últimos, sólo pueden aplicarse a *un* elemento de la página. Quiere decir que si hay un elemento que tiene asignado el atributo id="principal" no podrá haber otro id con igual valor (es decir, con el mismo nombre). La sintaxis en la CSS también es similar, solamente que en vez de usar un punto se utiliza el carácter de numeral (#):

```
h1#titulo1 {text-align: center}
#pregunta1 {color: green}
```

En el primer caso, aquel elemento h1 con id="titulo1" de la página será centrado. En el segundo, *cualquier* elemento que tenga asignado el atributo id="pregunta1" irá en verde.

Los selectores de id no tienen el grado de flexibilidad de los selectores de clases, por ejemplo, pero son una opción para aquellos casos en que se necesite identificar de modo exclusivo un determinado elemento de la página.

Selectores agrupados

Cuando tenemos varios selectores que comparten las mismas declaraciones pueden agruparse en una lista **separada por comas** (.). No se trata de un tipo de selector especial sino de una fórmula abreviada para aplicar las mismas propiedades a distintos selectores en un solo paso. Por ejemplo, si tenemos:

```
h1 {color:red}
p {color:red}
table {color:red}
#pregunta1 {color:red}
```

Podemos simplificarlo de este modo:

```
h1, p, table, #pregunta1 {color:red}
```

6. Selectores Genealógicos

Selectores de elementos descendientes

Los selectores de descendientes permiten especificar un elemento que está contenido dentro de otro elemento. Veamos el siguiente caso:

```
h1 {color: red}
p {color: black}
em {color: red}
```

Con estas reglas de estilo, el texto de los títulos sería rojo, el texto de los párrafos sería negro y el texto con énfasis también sería rojo. De este modo podemos destacar palabras dentro de un párrafo cambiando su color. Pero, nos resultaría imposible hacer lo mismo dentro de un título porque tanto **h1** como **em** son de color rojo.

Con un selector de descendiente podemos hacer que el elemento **em** sea de color azul

en los casos en que se encuentre dentro de un elemento **h1** (es decir, sea un descendiente de **h1**). La forma de especificar esta relación es usando ambos selectores (**h1** y **em**) separados por un espacio, poniendo en primer lugar el elemento padre:

```
h1 em {color: blue}
```

Con el agregado de este selector a la Hoja de Estilo, cuando destaquemos una palabra de un párrafo será de color rojo, pero si destacamos una palabra de un título será de color azul.

Si se deja un espacio entre dos elementos, se aplica el estilo al segundo elemento cuando sea descendiente del primero.

Si se quiere aplicar el estilo a todos los elementos, deben separarse por comas.

Selectores de hijos

Los selectores de hijos identifican a un elemento cuando es hijo de otro elemento. El signo ">" se utiliza para combinar los selectores y puede dejarse o no espacio alrededor del mismo:

```
DIV>P {color: blue}
```

```
DIV > P {color: blue}
```

El ejemplo anterior selecciona los párrafos que son hijos de un elemento DIV. Todas las consideraciones hechas para los selectores de descendientes se aplican en este caso

Selectores de primogénitos

La pseudo-clase `:first-child` selecciona un elemento que es el primer hijo de otro elemento. Por ejemplo, si queremos que un párrafo no tenga sangría cuando sea el primer hijo de un DIV:

```
DIV > p:first-child {text-indent: 0}
```

Entre P y `:first-child` no debe haber espacio, pero entre DIV y `>` y P pueden o no quedar espacios intermedios.

Si queremos seleccionar un elemento cuando sea el primer hijo de cualquier elemento, podemos usar el selector universal:

```
* > h1:first-child {color: blue} /* usando el asterisco */
h1:first-child {color: blue} /* o bien podemos obviar el *
```

Selectores de hermanos adyacentes

Los selectores de hermanos adyacentes tienen una sintaxis similar a los anteriores, usando como operador el signo más (+).

```
h1 + p {text-indent: 0}
h1 + h2 {margin-top: -0.5em}
```

En el primer caso, se selecciona los párrafos **p** que sigan inmediatamente a un título **h1** y se les quita la sangría. En el segundo, cuando a un título nivel 1 **h1** le sigue uno de nivel 2 **h2**, a este último se le adjudica un margen superior negativo para evitar que se separen demasiado entre sí.

Por hermanos adyacentes se entiende aquellos elementos que comparten un mismo padre y que se encuentran uno a continuación del otro en la estructura del documento. El elemento que se selecciona es el que se especifica en segundo lugar.

7. Pseudo-Selectores

Todos los selectores vistos hasta el momento se refieren a elementos que podemos encontrar dentro del código HTML. Las pseudo-clases (y los pseudo-elementos), por el contrario, no pueden deducirse simplemente observando la estructura del documento. Puede decirse que son abstracciones que permiten referirse a elementos que de otro modo resultarían inaccesibles.

Ya hemos visto un ejemplo de pseudo-clase: para referirnos al primer hijo de un elemento usábamos `:first-child`. El resto de pseudo-clases y pseudo-elementos tienen un tratamiento similar.

Las pseudo-clases clasifican a los elementos basándose en el estado del elemento, atributos que no puede deducirse de la estructura del documento. Las pseudo-clases pueden ser dinámicas, en el sentido de que un elemento puede adquirir o perder una pseudo-clase a medida que el usuario interactúa con el documento.

Pseudo selectores de enlaces

Los navegadores normalmente visualizan de forma diferente los enlaces (links) visitados de los no visitados. En CSS, esta visualización se puede definir a través de las pseudo-clases `:link` y `:visited`. De este modo podemos definir el estilo para los links que aparecen en nuestras páginas y otro estilo para esos links cuando ya han sido visitados. En HTML, dentro de la etiqueta `body` también se puede especificar un color para los links y otro para los links visitados, pero con CSS se pueden cambiar muchos otros atributos.

```
a:link { color: red; font-weight: bold; text-decoration:
a:visited { color: blue; }
```

En el ejemplo, los links sin visitar aparecen en color rojo, en negrita y sin subrayar, mientras que los visitados aparecen en azul.

Pseudo-clases dinámicas

Las pseudo-clases clasifican a los elementos basándose en el *estado* del elemento, atributos que no puede deducirse de la estructura del documento. Las pseudo-clases pueden ser **dinámicas**, en el sentido de que un **elemento puede adquirir o perder una pseudo-clase** a medida que el usuario **interactúa** con el documento.

Estas son las pseudo-clases dinámicas:

- `:hover` se aplica cuando el cursor del mouse señala el elemento.
- `:active` se aplica cuando el elemento es activado (por ejemplo, cuando el usuario presiona el botón del mouse).
- `:focus` se aplica cuando el elemento recibe el foco.

Uno de los mayores desvelos de los diseñadores de páginas Web han sido siempre los links y la manera de hacer que estos cambien de aspecto al ser señalados por el mouse. Con frecuencia, la solución consistía en utilizar dos imágenes que se cambiaban mediante JavaScript u otro lenguaje. Afortunadamente, con CSS podemos simplificar todo el proceso y obtener efectos muy variados en nuestros enlaces.

Normalmente, se utilizan las cuatro pseudo-clases siguientes para controlar el comportamiento de los links.

```
a:link {color: red}
a:visited {color: gray}
a:hover {color: blue}
a:active {color: fuchsia}
```

Aquí especificamos un color rojo para los links, un color gris para los ya visitados, un color azul cuando es señalado y un fucsia cuando es activado. Se debe tener en cuenta que,

por las reglas de cascada, el orden en que deben ser especificadas estas cuatro pseudo-clases para que funcionen correctamente es el que se ha usado en el ejemplo.

Pseudo-clases de idioma

En HTML es posible especificar el idioma que se usa en un elemento mediante el atributo `lang`. Por ejemplo, para identificar que un párrafo está en francés se usa:

```
<P lang:"fr">
```

El código para identificar el idioma consta generalmente de dos letras: "es" español, "en" inglés, "de" alemán, "fr" francés, etc. Mediante la pseudo-clase `:lang` podemos seleccionar elementos en base a su idioma.

Como cada idioma tiene sus propias convenciones con respecto al formato (uso de itálicas, sangrías o comillas, por ejemplo), esta pseudo-clase nos permite describir cómo debe aparecer un elemento según el idioma usado. La siguiente regla especifica el tipo de comillas que debe usar un elemento `q` en francés:

```
q:lang(fr) { quotes: '« ' ' »' }
```

Pseudo-elementos

En la impresión sobre papel, a menudo se estila que la primer letra o la primer línea de un párrafo (o ambas cosas a la vez) tenga un formato diferenciado. Los pseudo-elementos `:first-letter` y `:first-line` permiten conseguir el mismo efecto al seleccionar la primer letra o línea de un elemento.

Por otro lado, `:before` y `:after` permiten insertar un contenido antes o después de un elemento. Son llamados pseudo-elementos porque en realidad no existen en el documento fuente (ninguna marca identifica la primer letra de un párrafo, por ejemplo) pero son muy útiles para seleccionar elementos importantes dentro de la composición.

El pseudo-elemento `:first-line` permite aplicar un estilo determinado a la *primera línea* de un párrafo. Por ejemplo:

```
p:first-line { text-transform: uppercase }
```

Con la regla anterior conseguiríamos que todas las letras de la primera línea de los párrafos fueran convertidas en mayúsculas. Por supuesto, la medida de esta primer línea será determinada por el tamaño de la fuente, el ancho de la ventana, etc.

Sólo las siguientes propiedades se aplican al pseudo-elemento `:first-line`: propiedades de la fuente, propiedades del color, propiedades del fondo, 'word-spacing', 'letter-spacing', 'text-decoration', 'vertical-align', 'text-transform', 'line-height', 'text-shadow'

y 'clear'.

El pseudo-elemento **:first-letter** nos permite seleccionar la primer letra de un elemento (generalmente un párrafo o cabecera), normalmente para utilizarla como capitular (en un tamaño mayor que el resto del párrafo).

```
p { font-size: 12pt; line-height: 12pt }  
p:first-letter { font-size: 200%; font-weight: bold }
```

Las reglas anteriores determinan que los párrafos tengan una capitular en negritas del doble de tamaño que la fuente del párrafo.

Con los pseudo-elementos **:before** (*antes*) y **:after** (*después*) se puede insertar un contenido antes o después de un elemento determinado y definir el estilo del contenido insertado. La propiedad 'content', junto con estos pseudo-elementos, especifican lo que se inserta.

Podemos lograr que antes de cada elemento `<h3>` aparezca el texto "*Tema:*" sin necesidad de tener que escribirlo en cada título. También podemos hacer que cada párrafo termine con un pequeño icono o poner "*Fin*" al pie de cada página usando las siguientes reglas.

```
h3:before {content: "Tema: "}  
p:after {content: url("icono.gif")}  
body:after {content: "Fin"; display: block;}
```

En la última regla hemos especificado también "display: block" para que la palabra "*Fin*" comience en una nueva línea (como si fuese un nuevo párrafo).

8. Selectores de atributos

Los selectores de atributos permiten seleccionar elementos de la página según sus propiedades o el valor asignado a estas propiedades. Supongamos que hemos creado varias clases de párrafos y queremos identificar a *todos* aquellos que tengan especificado el atributo **class** (cualquiera sea el valor de ese atributo) para darle un margen izquierdo de 1 cm:

```
p[class] {margin-left: 1cm}
```

Ahora, si queremos seleccionar solamente aquellos párrafos que tengan el atributo `class="pregunta"`, debemos usar:

```
p[class="pregunta"] {margin-left: 1cm}
```

Podemos usar varios selectores de atributos para hacer aún más específica la regla. Supongamos que a ciertos párrafos con `class="pregunta"` también le hemos asignado el atributo `width` y queremos darle un margen izquierdo de 2 cm.

```
p[class="pregunta"][width] {margin-left: 2cm}
```

Hemos visto dos formas que pueden tomar estos selectores: `[atributo]` y `[atributo=valor]`. Adicionalmente, existen dos fórmulas para este tipo de selectores que se utilizan cuando a un determinado elemento se le han asignado una lista de valores en vez de un solo valor (más adelante en esta guía veremos los casos específicos).

`[atributo~=valor]` se usa para seleccionar los elementos que tienen como atributo una lista de palabras separadas por *espacios*, una de las cuales es exactamente valor.

`[atributo|=valor]` se usa para seleccionar los elementos que tienen como atributo una lista de palabras separadas por guiones, comenzando por valor.

Veamos un ejemplo de los dos casos:

```
img[alt~="logotipo"] {border: solid}
```

```
p[lang|="en"] {font-family: "Times New Roman", Serif}
```

El primero selecciona las imágenes cuyo atributo ALT tiene en su valor la palabra *logotipo*, como en `ALT="logotipo de la compañía"` o `alt="éste es nuestro logotipo"`. El segundo, selecciona los párrafos que tienen asignado como valor del atributo `lang` palabras que comienzan con *en*, como `en-US` o `en-cockney`.

Los profesores de este Curso WebCSS son:

A. Gámez, L.M. Marín, F. Mesa & S. Fandiño
