

Conceptos básicos para la definición de
estilos CSS

Conceptos Básicos de CSS

Bases de Datos 2012/2013

Luis Valencia Cabrera

Contenido

Reglas CSS.....	2
Selectores	3
Selectores básicos	3
Ejercicios.....	6
Selectores avanzados	6
Ejercicio	8
Pseudo-clases y pseudo-elementos	9
Pseudo-clases	9
Pseudo-elementos	10
Novedades en los selectores en CSS3:	10
Declaraciones	11
Consideraciones	11

Reglas CSS

- Las reglas CSS, como hemos visto, se componen de **selector** y **declaración**, estando la declaración compuesta por una serie de pares **propiedad: valor**.
- Como hicimos referencia al final del documento anterior, tenemos un buen tutorial introductorio aquí:
 - http://www.librosweb.es/css/capitulo_1/sintaxis_de_la_definicion_de_cada_propiedad_css.html.
- Iremos repasando los conceptos básicos acerca de los selectores, así como del tipo de propiedades y valores que podemos emplear para varios fines relacionados con el estilo.
- En la referencia de arriba se emplean algunas notaciones que conviene conocer para definir los valores de las propiedades, en su mayoría procedentes de la teoría acerca de **expresiones regulares**:
 - Si encontramos que algo se debe escribir como **valorA valorB**, separados por espacios sin ningún carácter especial, esto significa que se debe escribir tal cuál a la hora de escribir los valores para la propiedad.
 - Si tenemos elementos separados por barras verticales, |, como en el caso de **<porcentaje> | <medida> | inherit**, significa que el valor de la propiedad puede ser uno de los que se indica (los que están entre < y > serían un % válido o una medida válida, mientras que sin < y > significa que hay que ponerlo literal).
 - Si tenemos elementos separados por ||, se puede indicar uno o más de los valores indicados y en cualquier orden, como en el caso de: **<color> || <estilo> || <medida>**.
 - Si tenemos algo del tipo **expresión***, implica que podemos poner la expresión cero o más veces.
 - Si tenemos **expresión+**, implica que podemos ponerlo 1 o más veces.
 - Si tenemos **expresión?**, implica que se puede tener o no la expresión.
 - Si tenemos **expresión{valor_min,valor_max}**, implica que la expresión puede repetirse un número de veces dentro del rango dado por los extremos indicados.
 - Ejemplos:
 - **[<family-name> ,]*** → Pueden aparecer cero o más veces el par compuesto de un nombre de familia de fuente, seguido de una coma.
 - **<url>? <color>** → Puede ponerse una url o no, e irá seguida de un color, en este caso obligatorio.
 - **[<medida> | thick | thin] {1,4}** → Puede ponerse de 1 a 4 veces un elemento de los indicados: una medida, la palabra thick o la palabra thin.

Selectores

- Las declaraciones nos dirán qué estilo aplicar, pero antes debemos saber a qué elemento. Este es el papel que juegan los selectores, y como veremos permiten una gran variedad a la hora de seleccionar los elementos afectados por una regla.
- A un mismo elemento del DOM le pueden afectar varias reglas, y una regla puede afectar a varios elementos.
- Veamos a continuación los tipos básicos de selectores, ya que hay muchos más.

Selectores básicos

- **Selector universal:** el asterisco (*). El estilo en cuestión se aplica a todo elemento de la página.

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

- **Selector de tipo o etiqueta:** viene dado por el nombre de la etiqueta, y afecta a todos los elementos que tienen dicha etiqueta, como en estos ejemplos:

```
h1 { color: red; }      h2 { color: blue; }      p { color: black; }  
h1 { color: red; }      h2 { color: blue; }      p { color: black; }
```

Podemos indicar que un estilo se va a aplicar a varios selectores, en cuyo caso aparecerían separados por comas. Ello no impide que aspectos adicionales de cada elemento se puedan poner en reglas adicionales por separado.

```
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; }
```

- **Selector descendente:** indica que el estilo se aplicará a una etiqueta de un determinado tipo, siempre que ésta esté contenida en algún nivel dentro de otra de otro determinado tipo indicado (tenemos en cuenta el árbol del DOM).

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

Así, este estilo se aplicaría a la región marcada por cada etiqueta span contenida en el subárbol dependiente de alguna etiqueta p. En este ejemplo:

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

aplicaríamos el estilo tanto al texto 1 como al texto 2. Otros elementos span fuera de p no se verían afectados.

Se pueden anidar de la siguiente forma:

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

en cuyo caso se aplicaría el estilo a los elementos de tipo **em** (siempre se aplica al último elemento) que estén en algún nivel dentro de span, a su vez dentro de a, dentro de p.

Se pueden combinar también con el selector universal, como en este ejemplo:

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

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

¿A qué elementos se aplicaría el estilo indicado?

- **Selector de clase:** afecta a todos los elementos calificados dentro de una clase, con **class** dentro del código HTML.

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

Para indicar el estilo se debe anteponer un punto (.) al nombre de la clase empleado como selector:

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

Este estilo se aplicaría a todo elemento de la clase indicada:

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

Ahora bien, si únicamente queremos aplicarlos a los elementos con una determinada etiqueta que pertenecen a una clase, debemos poner el **etiqueta.clase** {...} sin ningún espacio ni coma, porque eso implicaría la aplicación de algunos de los selectores anteriores. Vemos la diferencia:

```
/* 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 { ... }
```

Un elemento en HTML puede pertenecer a varias clases, con declaraciones del tipo:

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

Esto implica la pertenencia a las clases especial, destacado y error, lo que permitiría poner condiciones sobre clases individuales o bien sobre varias de forma compuesta:

```
.error { color: red; }
.error.destacado { color: blue; }
.destacado { font-size: 15px; }
.especial { font-weight: bold; }
```

En este caso, la segunda regla se aplicaría sobre los elementos clasificados como destacado, siempre que estén calificados también como error, y no a los que estén solamente calificados con una de las 2 clases.

- **Selector individual:** se aplica únicamente al elemento de la página que tenga el id indicado. Se escribe anteponiendo una almohadilla (#) al id:

```
#destacado { color: red; }

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

El id es único dentro de la página. Sin embargo, podemos hacer algo del tipo:

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

Ya que con la importación de las hojas de estilo dentro de varias páginas podría darse el caso de tener los mismos ids, pero en distintos tipos de elementos. En ese caso, se aplicaría únicamente a aquellos que sean de tipo p.

Como en el caso anterior, no hay que confundir con selectores que incluyan espacios o comas:

```
/* 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 { ... }
```

Ejercicios

¿Qué elementos seleccionarían las combinaciones siguientes?

- .aviso .especial { ... }
- div.aviso span.especial { ... }
- ul#menuPrincipal li.destacado a#inicio { ... }

Realizar el ejercicio de la siguiente página:

- http://librosweb.es/css/capitulo_15.html

Selectores avanzados

- Esta información se ha tomado del documento:
 - http://www.librosweb.es/css/capitulo_2/selectores_avanzados.html
- Identificamos varios tipos de selectores avanzados que darán mucho juego.
- **Selectores de hijos:** se aplica el estilo a los elementos que sean *hijos directos* de otros. Es decir, no están a cualquier nivel dentro de una etiqueta, sino en el nivel inmediato.

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

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

En este caso, el estilo se aplicaría únicamente al primer span.

- **Selector adyacente:** se aplica el estilo a un elemento que esté al mismo nivel que otro, justo a continuación de él. Se indica con el símbolo +, como en la segunda regla del ejemplo:

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

Se aplicaría únicamente al elemento con etiqueta h2 a continuación de otro con etiqueta h1.

```
<body>
<h1>Titulo1</h1>

<h2>Subtítulo</h2>
...

<h2>Otro subtítulo</h2>
...
</body>
```

- **Selector hermano:** se aplica a todos los elementos al mismo nivel que el predecesor indicado, en lugar de únicamente el siguiente. Se indica con el símbolo ~.

Si quisiéramos aplicar el ejemplo anterior a todos los h2 dentro de body, a continuación del primer h1:

```
h1 ~ h2 { ... }
```

- **Selector de atributos:** permite seleccionar elementos en función del valor de sus atributos. Existen varios criterios de comparación para estos valores.
- En CSS2 son los siguientes:
 - **[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.

Algunos ejemplos:

```
/* 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] { ... }

/* 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 }
```

- En CSS3 aparecen algunos nuevos:
 - `[nombre_atributo^=valor]`, selecciona los elementos cuyo valor comienza con el valor indicado (ejemplo combinando con a: `a[src^="https"]`).
 - `[nombre_atributo$=valor]`, selecciona los elementos cuyo valor comienza con el valor indicado (elemento combinando con a: `a[src$=".pdf"]`).
 - `[nombre_atributo*=valor]`, selecciona los elementos cuyo valor contiene al valor indicado (elemento combinando con a: `a[src*="w3schools"]`).

Más ejemplos en:

- <http://css-tricks.com/attribute-selectors/>

Hay muchos más selectores, que pueden consultarse en la URL:

- http://www.w3schools.com/cssref/css_selectors.asp

Se puede comprobar la compatibilidad de nuestro navegador con los distintos tipos de selectores en la siguiente dirección:

- <http://tools.css3.info/selectors-test/test.html>

Ejercicio

- Realizar el ejercicio de la página:
 - http://librosweb.es/css/capitulo_15/ejercicio_2.html

Pseudo-clases y pseudo-elementos

- Vienen a complementar a los selectores anteriores con criterios adicionales.
- Se suelen indicar con :, seguidos de la pseudo-clase correspondiente.
- Como veremos, hay una amplia gama (más detallada en http://www.librosweb.es/css_avanzado/capitulo_3/pseudo_clases.html).

Pseudo-clases

- La pseudo-clase **:first-child**: selecciona el primer elemento hijo de un elemento.

```
p em:first-child {  
    color: red;  
}
```

```
<p>Lorem <span><em>ipsum dolor</em></span> sit amet, consectetur adipiscing  
elit. Praesent odio sem, tempor quis, <em>auctor eu</em>, tempus at, enim  
. Praesent nulla ante, <em>ultrices</em> id, porttitor ut, pulvinar quis,  
dui.</p>
```

Este estilo se aplicaría al primer **em** hijo de cada **p**. También se podría incluir un selector para el primer **em** hijo de cualquier elemento:

```
p:first-child { ... }
```

- Las pseudo-clases **:link** y **:visited**: se pueden utilizar para aplicar diferentes estilos a los enlaces de una misma página:
 - La pseudo-clase **:link** se aplica a todos los enlaces que todavía no han sido visitados por el usuario.
 - La pseudo-clase **:visited** se aplica a todos los enlaces que han sido visitados al menos una vez por el usuario.

```
a:link    { color: red;  }  
a:visited { color: green; }
```

- Las pseudo-clases **:hover**, **:active** y **:focus**: permiten al diseñador web variar los estilos de un elemento en respuesta a las acciones del usuario. Al contrario que las pseudo-clases **:link** y **:visited** que sólo se pueden aplicar a los enlaces, estas pseudo-clases se pueden aplicar a cualquier elemento.
 - **:hover**, se activa cuando el usuario pasa el ratón o cualquier otro elemento apuntador por encima de un elemento.
 - **:active**, se activa cuando el usuario activa un elemento, por ejemplo cuando pulsa con el ratón sobre un elemento. El estilo se aplica durante un espacio de tiempo prácticamente imperceptible, ya que sólo dura desde que el usuario pulsa el botón del ratón hasta que lo suelta.
 - **:focus**, se activa cuando el elemento tiene el foco del navegador, es decir, cuando el elemento está seleccionado. Normalmente se aplica a los elementos `<input>` de los formularios cuando están activados y por tanto, se puede escribir directamente en esos campos.

El orden entre estos operadores es importante, luego si se aplican varios han de hacerse en el orden:

1. `a:link { ... }`
2. `a:visited { ... }`
3. `a:hover { ... }`
4. `a:active { ... }`

Pseudo-elementos

- El **pseudo-elemento** *:first-line*: permite seleccionar la primera línea de texto de un elemento. Así, la siguiente regla CSS muestra en mayúsculas la primera línea de cada párrafo:

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

Este pseudo-elemento sólo se puede utilizar con los elementos de bloque y las celdas de datos de las tablas. Se pueden combinar, como en:

```
div:first-line { color: red; }
p:first-line { text-transform: uppercase; }

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

- El **pseudo-elemento** *:first-letter*: El pseudo-elemento `:first-letter` permite seleccionar la primera letra de la primera línea de texto de un elemento. De esta forma, la siguiente regla CSS muestra en mayúsculas la primera letra del texto de cada párrafo:

```
p:first-letter { text-transform: uppercase; }
```

- Los **pseudo-elementos** *:before* y *:after*: Los pseudo-elementos `:before` y `:after` se utilizan en combinación con la propiedad `content` de CSS para añadir contenidos antes o después del contenido original de un elemento.

Las siguientes reglas CSS añaden el texto Capítulo - delante de cada título de sección `<h1>` y el carácter . detrás de cada párrafo de la página:

```
h1:before { content: "Capítulo - "; }
p:after { content: "."; }
```

El contenido insertado mediante los pseudo-elementos `:before` y `:after` se tiene en cuenta en los otros pseudo-elementos `:first-line` y `:first-letter`.

Novedades en los selectores en CSS3:

- Además de los selectores ya introducidos (destacando *not*, *nth-child* y *empty*):
 - http://www.librosweb.es/css_avanzado/capitulo_3/selectores_de_css_3.html

Declaraciones

- Hay una gran variedad de posibles declaraciones de propiedades de atributos, que podemos agrupar en bloques semánticos.
- Una descripción muy buena se encuentra en el curso web, del capítulo 3 en adelante:
 - http://librosweb.es/css/capitulo_3.html
- Resulta especialmente importante e interesante la explicación del modelo de cajas ([capítulo 4](#)). Echemos un vistazo y, finalmente, realicemos el ejercicio que aparece en [este enlace](#).

Consideraciones

- Sobre prioridad de aplicación de hojas de estilo:
 - Consultar: http://www.librosweb.es/css_avanzado/capitulo_2.html
- En general, técnicas avanzadas además de lo que aparece en el tutorial anterior:
 - http://www.librosweb.es/css_avanzado/
- Referencia detallada:
 - <http://www.librosweb.es/referencia/css/>
 - <http://www.w3schools.com/cssref/default.asp>