Sintaxis CSS y Selectores

Las Reglas en CSS3

Nuestro código **CSS3** no es más que una serie de **reglas** incluidas en un fichero. La mejor forma de entender una regla es con un ejemplo:

```
selector {
  declaracion [propiedad: valor;]
}
```

Como puedes observar, **una regla se compone de un selector y una declaración** que está encerrada entre {} de forma que con la declaración decimos *qué tiene que hacerse* y es el selector el encargado de indicar a*qué elemento* de nuestro HTML se le debe aplicar la declaración.

La declaración es un conjunto de líneas separadas por ; (punto y coma) y cada línea (aunque pueden estar todas seguidas) se compone de una propiedad y un valor de forma que especificamos **qué modificación visual queremos realizar (propiedad) y en qué medida (valor)**.

Los Selectores en CSS3

Hasta ahora hemos utilizado exclusivamente los llamados selectores de tipo o de etiqueta y los selectores descendentes, sin embargo tenemos varios tipos de selectores:

- Selector universal.
- Selector de tipo o etiqueta.
- Selector descendente.
- Selector de clase.
- Selector de ID.
- Selector de hijos.
- Selector de atributos.

Selector universal

Es muy poco utilizado, ya que aplicará el estilo a todas las etiquetas del documento. Sin embargo, es muy interesante a la hora de establecer una serie de valores básicos (como puede ser el color de la fuente, la familia y tamaño) de forma que se garantice un estilo uniforme y sea en reglas más concretas dónde se apliquen los estilos específicos.

En el **CSS** de nuestro proyecto, podríamos utilizar el selector universal en la primera regla declarada, por lo que en vez de:

```
html, body, div, span, applet, object, iframe, h1, h2, h3, h4, h5, h6, p, blockquote,
pre, a, abbr, acronym, address, big, cite, code, del, dfn, em, img, ins, kbd, q, s,
samp, small, strike, strong, sub, sup, tt, var, b, u, i, center, dl, dt, dd, ol, ul,
li, fieldset, form, label, legend, table, caption, tbody, tfoot, thead, tr, th, td,
article, aside, canvas, details, embed, figure, figcaption, footer, header, hgroup,
menu, nav, output, ruby, section, summary, time, mark, audio, video {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  vertical-align: baseline;
}
```

lo mejor sería:

```
* {
  margin: 0;
  padding: 0;
  border: 0;
  font-size: 100%;
  vertical-align: baseline;
}
```

Selector de tipo o etiqueta

Como su propio nombre indica, el selector de etiqueta se utiliza para aplicar estilos en las **etiquetas HTML**, es lo que hemos estado realizando constantemente:

etiqueta_html {
  propiedad: valor;
}

Selector descendente

El otro de los selectores que utilizamos en nuestro proyecto es el **selector descendente**. Nos permite establecer un estilo para una etiqueta que se encuentre dentro de otra, pudiendo tener un estilo para las etiquetas *h2* que están dentro de un *article* y otro estilo para los *h2* que están dentro de un *aside*

etiqueta_html etiqueta_html {
  propiedad: valor;
}

Podría asegurarse que es el selector más utilizado y nos proporciona una enorme flexibilidad ya que este selector nos permite tener **diferentes estilos para las mismas etiquetas HTML** según su jerarquía dentro del documento

```css
article p {
  text-align: justify;
}
/* En esta regla utilizamos un selector de etiquetas para indicar que el texto dentro de la etiqueta HTML p ubicada dentro de un article se debe pintar justificado */
aside p {
  text-align: right;
}
/* En esta regla utilizamos un selector de etiquetas para indicar que el texto dentro de la etiqueta HTML p ubicada dentro de un aside se debe pintar alineado a la derecha */
```

Selector de clase

El selector de clase es, sin duda, el rey de los selectores. A las etiquetas HTML podemos ponerle un atributo*class* y asignarle un valor:

  o bien

El selector de clase tiene una peculiaridad a la hora de introducirlo en la regla CSS y es que su nombre va precedido por un . (punto) por lo que en nuestro ejemplo sería:

```css
.clase1 {
  propiedad: valor;
}

.clase2 {
  propiedad: valor;
}
```

También podemos combinar un selector de etiqueta con un selector de clase:

```css
img.clase1 {
  propiedad: valor;
}
```

Selector de ID

Al igual que una etiqueta HTML tiene un atributo *class*, existe otro llamado *id*. Al atributo *id* sólo se le puede asignar un valor (no como en el *class* que podemos utilizar varios) y, además, **dentro de un documento HTML no pueden existir dos (o más) etiquetas con el mismo *id***. El *id* es, como su propio indica, un identificador… y los identificadores tienen que ser únicos!!

IMPORTANTE: Si utilizas el mismo *id* en varias etiquetas dentro del mismo documento, tu navegador web aplicará los estilos correspondientes a esas etiquetas, pero cuando pases tu código por el validador de la W3C te indicará que existe un error ya que el estándar web especifica que el *id* es único y tenemos que seguir (siempre que podamos) los estándares… si te ves en la situación de tener que repetir un *id*… lo suyo es utilizar una clase. Es más!!! Las tendencias actuales de buenas prácticas nos dictan que NO debemos utilizar los IDs en las hojas de estilo, dejando éstos para su uso con JavaScript.

Dicho esto, al igual que a la hora de incorporar nuestra clase a la regla CSS la precedemos por un punto, el ID viene precedido por una almohadilla #

```
#identificador {
  propiedad: valor;
}
/* Esta regla se aplica a una etiqueta HTML con id igual a identificador, por ejemplo
<div id="identificador"></div> */
```

Podemos combinar nuestro selector ID con los selectores de etiquetas, descendentes e incluso de clase… aunque si necesitas hacerlo deberías plantearte si un selector ID es lo más correcto… la respuesta siempre será no

Selector de hijos

El selector de hijos es muy parecido al descendente, pero con una importantísima diferencia. Un selector descendente aplica un estilo a una etiqueta (clase, etc) que está contenida en otra, esto significa que este estilo:

```
div img {
  border: 0;
}
```

se aplicaría para la etiqueta *img* en estos tres casos:

```
<div>
  <img src="url imagen" alt="texto alternativo" title="titulo imagen" />
  <p>
    <img src="url imagen2" alt="texto alternativo2" title="titulo imagen2" />
  </p>
  <img src="url imagen3" alt="texto alternativo3" title="titulo imagen3" />
</div>
```