

### 1. Introducción a los selectores

Para poder hacer el diseño de la página existen los mecanismos denominados reglas, selectores y declaraciones. Cada regla o conjunto de reglas consiste en uno o más selectores y un bloque de estilo o declaración. En el interior de estos bloques se definen elementos de aplicación, sobre los cuales se da valor a las propiedades deseadas. Los estilos se aplican a los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con la sintaxis. La declaración indica “lo que hay que hacer”, y el selector indica “sobre qué elemento hay que hacerlo”. A un mismo elemento HTML se le pueden asignar infinitas reglas CSS, y a cada regla CSS puede aplicarse a un número infinito de elementos. Una misma regla puede aplicarse sobre varios selectores y un mismo selector se puede utilizar en varias reglas.

```
body{
  background-color:olive;
  color:white;
}
```

La estructura siempre es la misma:

1. Se indica el **selector** (para saber a quién afecta el código), que en este caso es **body**. La línea se finaliza con una **llave {**.
2. Posteriormente se indica la **propiedad** a modificar (en este caso **background-color** para modificar el color de fondo y **color** para especificar el color del texto).
3. Seguido a la propiedad se indica el **valor** (separados ambos por dos puntos (:).
4. Para finalizar cada una de las sentencias CSS se utiliza obligatoriamente el símbolo **(;)** **punto y coma**.
5. Para terminar el bloque CSS utilizamos la **llave }**.

```
selector{
  propiedad:valor;
  propiedad:valor;
}
```

### 2. Selectores básicos.

Hay múltiples selectores y en función del rango de acción sobre el que se modifican las propiedades de los elementos que se definen en su declaración encontramos el selector **universal**, **selector de etiqueta**, **selector descendente**, etc...

#### 2.1 Selector universal

Permite seleccionar todos los elementos de una misma página, dando el mismo formato a todos los elementos recogidos en la página y que haga uso de la hoja de estilo CSS.

*Sintaxis del selector universal:*

```
*{
  propiedad:valor;
}
```

```
*{
  font-size: 3em;
  color: red;
  text-align: left;
}
```

## 2.2 Selector etiqueta

Permite seleccionar todos los elementos que contengan la etiqueta indicada en dicho selector y se les establecerá un mismo formato.

*Sintaxis del selector etiqueta:*

```
etiquetaHTML{  
    propiedad:valor;  
}
```

```
h1{  
    font-size: 3em;  
    color: red;  
    text-align: left;  
}  
  
h2{  
    font-size: 2em;  
    color: blue;  
    text-align: center;  
}
```

En el ejemplo se diferencian estilos para las etiquetas <h1> y <h2>. Los elementos de h1 estarán alineados a la izquierda y serán de color rojo, mientras que a los de tipo h2 se les aplica una alineación central y son de color azul.

La creación de los selectores puede realizarse de varias formas:

-**Selectores individuales** que proporcionan estilo a un tipo de elemento.

-**Selectores encadenados** que definen el mismo estilo para todos los elementos detallados en el selector.

-**Selectores anidados**, parecidos a los encadenados pero añadiendo características individuales para algunos elementos

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

```
h1,h2,h3{  
    color: #8A8E27;  
    font-weight:normal;  
    font-family: Arial, Helvetica, sans-serif;  
}
```

Selectores encadenados: los elementos van separados por una coma (,) y se aplican los mismos estilos a todos

Selectores individuales

```
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;  
}
```

Selectores anidados: se indican los encadenados y después se especifican las particularidades.

## 2.3 Selector descendente

Un elemento es descendente de otro cuando se encuentra entre las etiquetas de apertura y de cierre del elemento “padre”. Este tipo de selector seleccionará los elementos del tipo especificado que se encuentra dentro de otro elemento, también denotado en la sintaxis del selector. Es decir, las propiedades recogidas en la declaración se aplicarán solo a los elementos de la etiqueta citada que estén dentro de otra etiqueta concreta.

Sintaxis:

```
selector1 selector2 ....selectorN{  
    propiedad:valor;  
}
```

## Selectores, clases e identificadores

En el siguiente ejemplo se aplica a los dos enlaces puesto que ambos se localizan bajo un selector de tipo <p>

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

Hay que tener en cuenta que el selector descendiente solo se aplica al último de ellos; el resto indican el lugar exacto en el que debe encontrarse el selector que va a recibir el estilo.

Otro ejemplo:

```
p a span em{  
  color: #001100;  
}
```

### 3. Selectores de clase e identificadores.

Hay ocasiones en las que, a pesar de pertenecer al mismo tipo elemento, no es deseable que presenten el mismo estilo. Para ello se utilizan los **selectores de tipo clase o tipo identificador**. A grandes rasgos, ambos distinguen usando un atributo identificador o de clase de la etiqueta HTML, a través de la cual se seleccionarán los elementos y se les aplicará el estilo oportuno.

#### 3.1 Identificadores (ID).

Seleccionan un atributo al que dar formato buscando en el contenido de los atributos "id" del documento estructurado en HTML. La llamada al identificador desde la hoja de estilo se hace precedida del carácter "#".

Sintaxis:

```
#identificador{  
  propiedad: valor;  
}  
----->  
<etiqueta id = identificador>  
  ...  
</etiqueta>
```

- Código HTML:

```
<h1 id="anexos">  
  Anexos 1  
</h1>  
<h2 id="anexos">  
  Anexos 2  
</h2>  
<h2 id="textos">  
  Textos  
</h2>
```

- Código CSS:

```
#anexos {  
  font-size: 18px; text-align: center;  
}
```

En este ejemplo el identificador queda definido de forma general para todas las etiquetas identificadas con id "anexos".

En el siguiente ejemplo tenemos un identificador dependiente que se aplica a todos los elementos de tipo div identificados con id="anexos":

```
div#anexos{  
    font-size: 18px; text-align: center;  
}
```

### 3.2 Clases (class).

Los selectores de clase modifican y dan estilo solo a aquellos elementos agrupados bajo un atributo de tipo **"class"** en el código HTML, y cuyo nombre de clase debe coincidir con el nombre de la clase definida en ese atributo. El nombre de la clase buscada debe ser exactamente el mismo nombre contenido en el atributo **class** en HTML. Se utiliza para agrupar a un conjunto de etiquetas que forman parte del mismo grupo o clase, aunque nos sean del mismo tipo.

Sintaxis:

```
.clase{  
    propiedad: valor;  
}  
  
-----> <etiqueta class=clase>  
           ...  
           </etiqueta>
```

- Código HTML:

```
<h1 class="clase1">  
    Esta es la clase 1.  
</h1>  
<h1 class="clase2">  
    Esta es la clase 2.  
</h1>
```

- Código CSS:

```
h1.clase1{  
    background-color: blue;  
}
```

### 3.3 Combinación de clases e identificadores.

Existe la posibilidad de aplicar estilos, al mismo tiempo, a elementos identificados mediante clases e identificadores; para ello basta con añadir al elemento un parámetro de cada tipo **"class"** e **"id"**, de tal forma que el elemento tendrá el estilo definido para la clase y para el identificador.

- Código HTML:

```
<h1 id="destacado" class="clase1">  
    Título.  
</h1>  
<h1 class="clase1">  
    Subtítulo.  
</h1>  
<h1 class="clase2">  
    Texto.  
</h1>
```

- Código CSS:

```
h1.clase1{  
    background-color: blue;  
}  
#destacado {  
    font-size: 18px; text-align: center;  
}
```

En este ejemplo, el primer elemento queda definido por un identificador y por una clase. Por otro lado el segundo solo está identificado por un atributo de clase. Desde la hoja de estilos,

todos aquellos elementos de tipo h1 definidos bajo la clase1, aparecerán con un fondo de color azul; así mismo, aquellos elementos que, además, están identificados con el id="destacado", se mostrarán con el estilo marcado en el selector #destacado; con tamaño de 18px y estarán centrados.

#### 4. Pseudoselectores: pseudoclases y pseudoelementos.

Las **pseudoclases** se utilizan para modificar el estilo de un elemento que puede presentar varios estados.

Los **pseudoelementos** se utilizan para modificar los estilos de determinados elementos especiales que no pueden ser modificados a través de otros selectores ni pseudoclases.

##### 4.1 Pseudoclases.

Hay elementos que pueden presentar más de un estado. En el diseño web, es interesante que, en función del estado en el que se encuentre el elemento, se presente un estilo u otro. Por ejemplo, en el caso de los enlaces, es común que estos cambien su apariencia en función de si han sido visitados o no. Por lo tanto podrían definirse pseudoclases como aquellos selectores utilizados para definir las propiedades de los elementos con diferentes estados.

Para el caso de la etiqueta <a> o enlace son posibles las siguientes pseudoclases en función de sus estados:

- a:link** -Enlaces que no han sido visitados por el usuario.
- a:visited** -Enlaces visitados al menos una vez por el usuario.
- a:hover** -El elemento se activa, modifica su estilo, cuando cualquier elemento apuntador pasa por encima de dicho elemento.
- a:active** -Este estado se activa cuando el usuario activa un elemento, normalmente, al pulsar con el ratón sobre el elemento.
- a:focus** -El elemento se activa cuando tiene el foco del navegador sobre él, por ejemplo, al seleccionar un elemento en un formulario.

Ejemplo:

- Código HTML:

```
<a href="#"> <h1>  
  Enlace  
</h1></a>
```

- Código CSS:

```
a:link {  
  font-family: sans-serif;  
}  
a:hover {  
  color:red;  
}  
a:visited {  
  color:gray;  
}
```

Otra de las pseudoclases más comunes es la relativa a la posición de un elemento en una lista o grupo de elementos hermanos. En estos casos, en función de su lugar en el listado tomará un valor u otro. Podemos encontrar algunas pseudoclases como las siguientes en función de los estados:

- :first-child** -Selecciona el primer elemento de un grupo de elementos.
- :first-of-type** -Se activa si es el primer elemento de la lista.

- :last-of-type** -Se activa si es el último elemento de la lista
- :only.child** -Se activa si es el único elemento de la lista.
- :empty** -Se activa si la lista está vacía.
- :nth-child** -Permite seleccionar varios elementos indicando su posición

Ejemplo:

- Código HTML:

```
<ul>
  <li>Nombre y cantidad</li>
  <li>Pepe. 5</li>
  <li>María. 8</li>
  <li>Manuel. 3</li>
  <li>Total. 16</li>
</ul>
```
- Código CSS:

```
li:first-child {
  font-size:22px; color:#333399; list-style-type: none;
}
li:last-of-type {
  font-size:16px; color:#333399 ; font-weight:bolder;
}
```

## 4.2 Pseudoelementos.

Permiten modificar el estilo de determinados elementos especiales, que no pueden ser modificados a través de selectores ni de pseudoclases. Normalmente, no describen un estado, sino que añaden estilos a una parte concreta del documento. Algunos pseudoelementos comunes son **first-line**, **first-letter**, **before** y **after**. De forma general su sintaxis es:

```
selector::pseudo-elemento {
  propiedad:valor;
}
```

- ::first-line** -Selecciona la primera línea de texto de un elemento.
- ::first-letter** -Selecciona la primera letra de la primera línea de un elemento.
- ::before** y **::after** -Se utilizan para añadir texto al elemento original. En el caso de **::before** se añade al principio y en el caso de **::after** se añade al final. En la propiedad **content** se indica el nuevo contenido que se va a incorporar.

Ejemplos:

```
p::first-line {
  color:red;
}

p::first-letter {
  color:green;
}

h1::before {
  content: "INICIO.-";
}

h1::after {
  content: ".-FIN";
}
```