

TEMA 4: INTRODUCCIÓN A CSS

4.1 FUNCIONAMIENTO DE LOS ESTILOS CSS

Los estilos CSS son un conjunto de normas que indican cómo el navegador debe visualizar cada uno de los diferentes elementos que contiene una página web. Aunque estas normas habitualmente están relacionadas con aspectos visuales y consideraciones decorativas, con CSS3 cada vez existen más posibilidades (animaciones, transiciones, eventos..).

Una de las utilidades más evidentes de CSS es la posibilidad de aplicar un estilo a múltiples páginas html, no teniendo que repetir los mismos estilos cada vez que creamos un html nuevo. En este caso se podría utilizar un único fichero CSS externo que afectaría al mismo tiempo a múltiples páginas html. También sería posible incluir el código CSS dentro del mismo fichero html, o bien añadir código CSS señalando sólo una parte del texto dentro del mismo html o incluso utilizar todas estas tres posibilidades conjuntamente.

Vamos a suponer que queremos que el fondo de nuestra página web tenga un bonito color verde oliva (nombre de color en HTML5: 'olive') y que además el color del texto sea blanco. Y para ello vamos a hacerlo de tres maneras distintas.

4.2 TRES MODOS DE UTILIZACIÓN

4.2.1. Utilizar un fichero CSS externo

Esta es una de las opciones más comunes, ya que posteriormente es posible utilizar el mismo fichero CSS para aplicar el mismo estilo a otras páginas html. De esta manera, cualquier cambio realizado en el diseño afectaría al mismo tiempo a todas las páginas html que utilicen dicho fichero CSS.

En este caso, por una parte tendríamos un fichero HTML y un fichero CSS, para conectar ambos debemos añadir el siguiente código en fichero HTML dentro del <head>.

```
<link rel="stylesheet" href="estilo.css" type="text/css"/>
```

En el ejemplo anterior, el fichero CSS se llamaría estilo.css y estaría ubicado en la misma carpeta que el html. Como lo lógico es que el fichero esté en una carpeta llamada css, lo único que tendríamos que hacer es añadir el nombre de la carpeta antes del fichero:

```
<link rel="stylesheet" href="css/estilo.css" type="text/css"/>
```

Dentro del fichero **estilo.css** tendríamos únicamente el siguiente contenido:

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

La estructura siempre es la misma:

- Primero se indica el selector (para saber a quién afecta el código) y la línea se finaliza la línea con un "{".
- Posteriormente se indica el atributo a modificar (en este caso 'background-color' para el color de fondo y 'color' para el color del texto).
- Seguido al atributo se indica el nuevo valor (separados ambos por dos puntos ":").
- Para finalizar los atributos para este selector, se cierra con un "}".

```
selector{  
  atributo:valor;  
  atributo:valor;  
}
```

4.2.2. Incluir el código CSS dentro del HTML

En este caso, como aspecto positivo, podemos decir que no necesitamos tener 2 ficheros, ya que todo está en el fichero html. Como punto negativo (o no) es que dicho estilo únicamente afectará al fichero html en el que se coloque el código CSS.



Este código se añade dentro del apartado '<head>', y debe estar dentro de estas 2 líneas:

```
<style type="text/css">
</style>
```

Siguiendo el ejemplo anterior, el código completo quedaría de la siguiente manera:

```
<style type="text/css">
  body{
    background-color:olive;
    color:white;
  }
</style>
```

4.2.3. Aplicar CSS a un trozo de texto muy concreto

Si lo que queremos es aplicar un estilo a un párrafo, a un frase o incluso a una palabra o letra en concreto insertaremos el código CSS dentro de un '' o '<p>' que encerrará el párrafo, frase, palabra o letra afectada:

Si dicho texto tiene aplicado otro estilo a través de un CSS externo o colocado en el html se aplicarán ambos estilos al mismo tiempo.

(El texto coloreado de color rojo es aquel que se mostrará en pantalla con los atributos que se le hayan asignado, éstos en color negro).

```
Texto <span style="color:red;font-size:40px">rojo y grande</span> y este
otro <span style="color:blue;
font-weight:bold;font-size:16px">azul, negrita y pequeño</span>
</style>
```

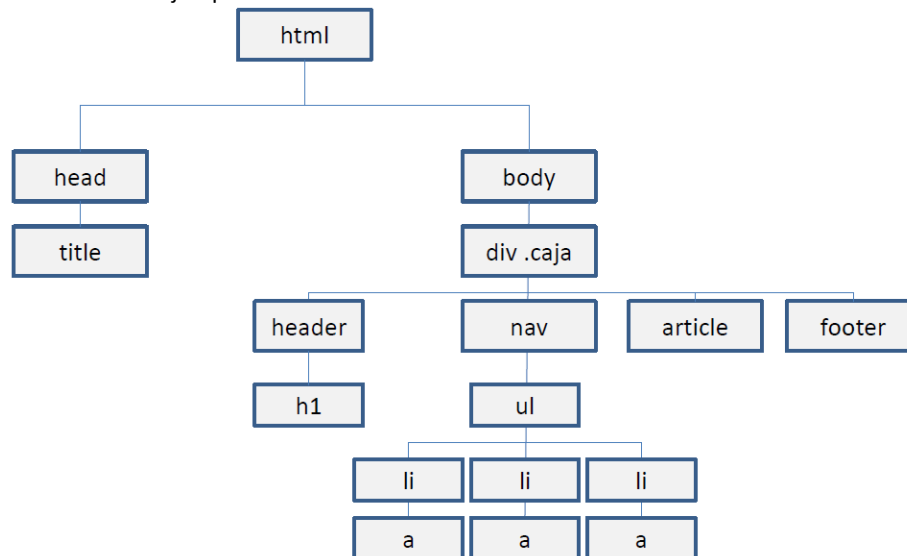
Texto **rojo y grande** y este otro azul, negrita y pequeño

4.3 HERENCIA CSS

Para entender cómo funcionan los selectores y la herencia CSS es necesario entender qué es el árbol del documento.

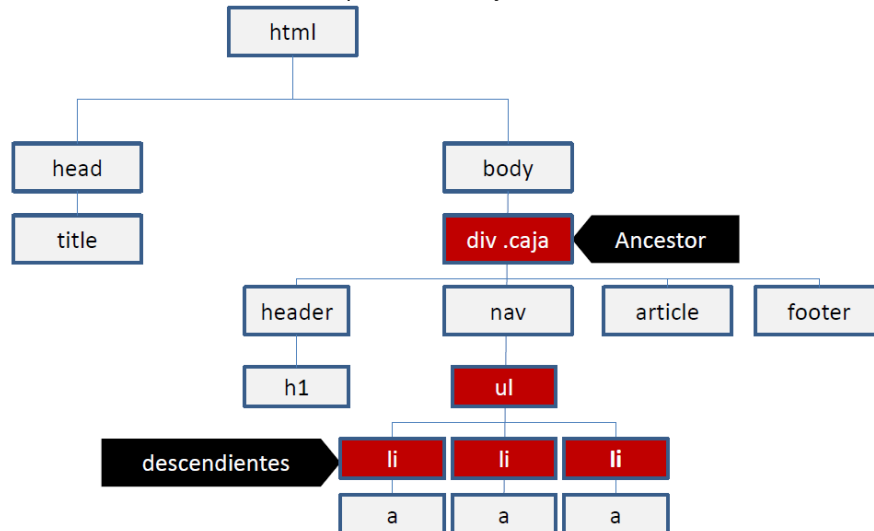
```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <link href="html5_estilo2.css" rel="stylesheet" />
6 <title>Árbol</title>
7 </head>
8 <body>
9 <div class="caja">
10   <header>
11     <h1>Cabecera: header</h1>
12   </header>
13   <nav>
14     <ul>
15       <li><a href="#">Menú 1</a></li>
16       <li><a href="#">Menú 2</a></li>
17       <li><a href="#">Menú 3</a></li>
18     </ul>
19   </nav>
20   <article>Contenido de la página</article>
21   <footer>Pie de página</footer>
22 </div>
23 </body>
24 </html>
```

El árbol del documento del ejemplo anterior:

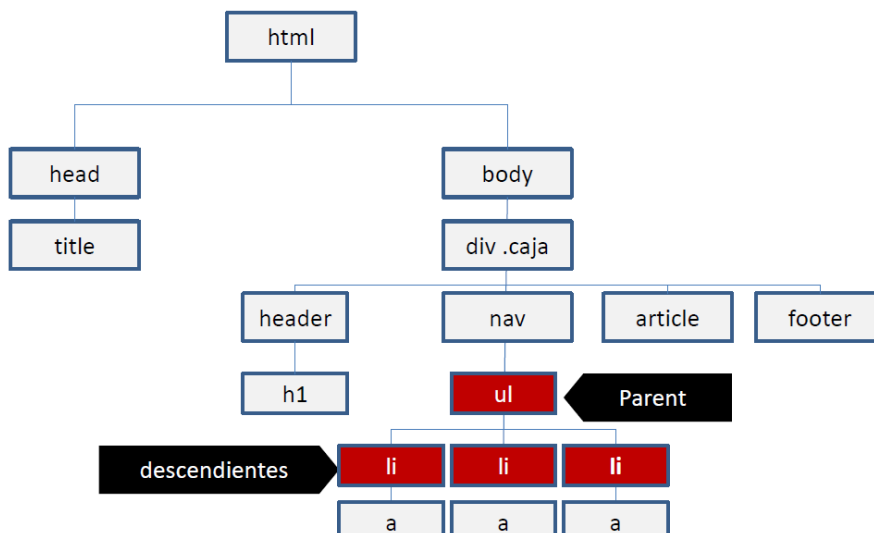


Ancestor es un elemento conectado pero más arriba en la estructura del documento

Descendientes son los elementos conectados pero más abajo en la estructura del documento.

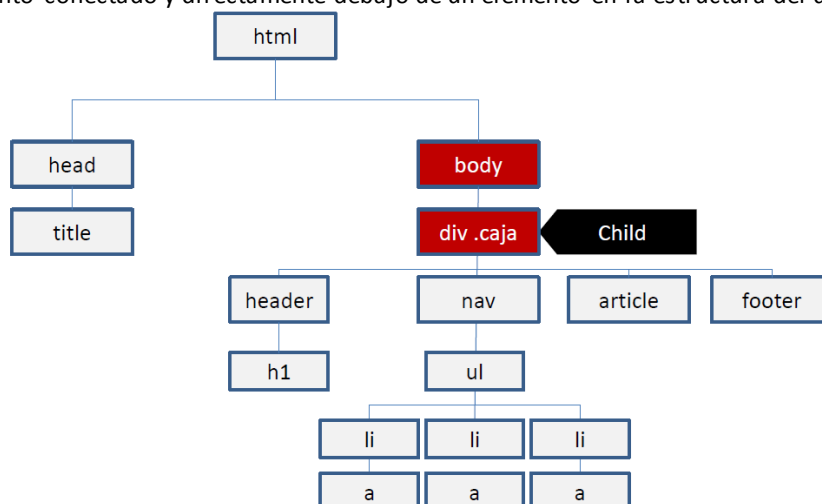


Parent es el elemento conectado y directamente sobre un elemento en la estructura del documento.

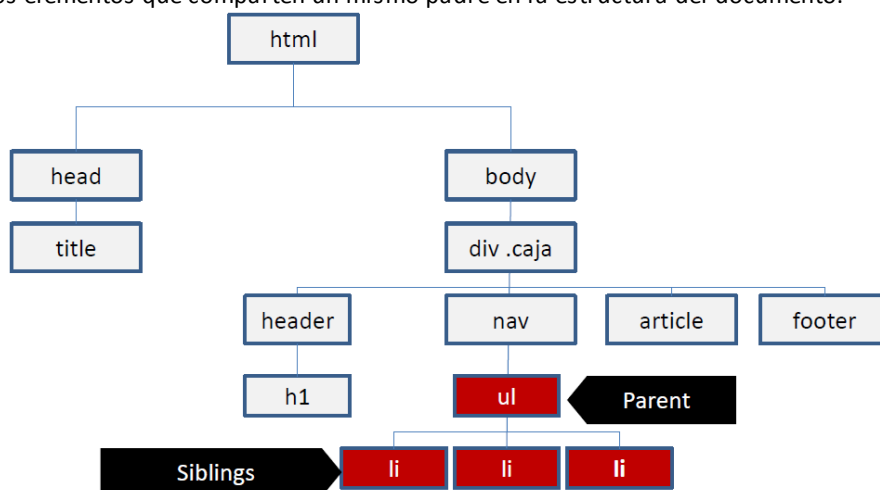




Child es el elemento conectado y directamente debajo de un elemento en la estructura del documento.

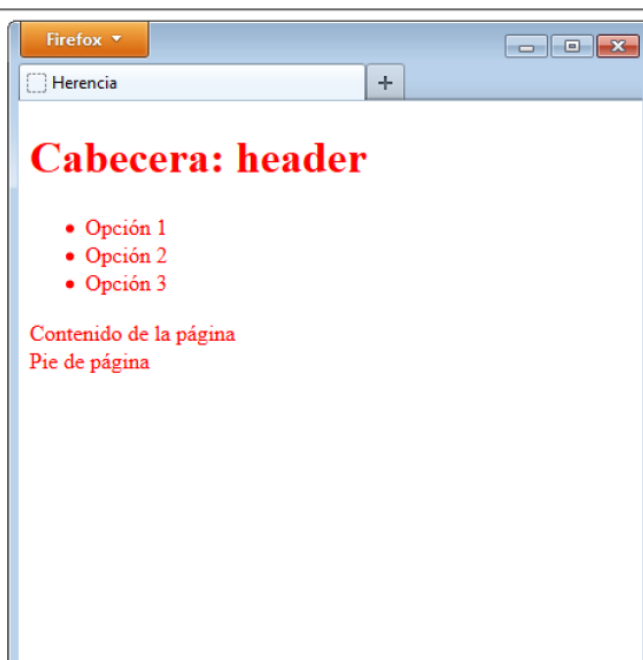


Siblings son los elementos que comparten un mismo padre en la estructura del documento.



Si definimos un estilo en el **body**, todos los elementos situados debajo en el árbol del documento, heredan esa propiedad.

```
1 <!doctype html>
2 <html>
3 <head>
4 <meta charset="utf-8">
5 <title>Herencia</title>
6 <style type="text/css">
7 body {
8     color: red;
9 }
10 </style>
11 </head>
12 <body>
13 <div class="caja">
14     <header>
15         <h1>Cabecera: header</h1>
16     </header>
17     <nav>
18         <ul>
19             <li>Opción 1</li>
20             <li>Opción 2</li>
21             <li>Opción 3</li>
22         </ul>
23     </nav>
24     <article>Contenido de la página </article>
25     <footer>Pie de página</footer>
26 </div>
27 </body>
28 </html>
```





4.3.1 Selectores y herencia

La palabra CSS viene de *Cascading StyleSheets*, esto quiere decir: Hojas de Estilo en Cascada. La palabra **cascada** hace referencia a una propiedad **muy importante** de las hojas de estilo, y es que los estilos aplicados a un elemento padre son heredados por su hijo.

Por ejemplo:

```
<ul style="color:red">
  <li>Inicio</li>
  <li style="color:blue">Experiencia</li>
</ul>
```

En este caso el color de la fuente "Inicio" aparecerá en rojo y la de "Experiencia" en azul.

Nota: te recuerdo que **no está recomendado** aplicar los estilos usando atributos. En este caso lo he hecho así porque creo que queda más clara la explicación.

Del mismo modo y como ya adelantábamos al principio del capítulo, si un estilo se define dos veces, el último definido será el que prevalecerá, por ejemplo si en nuestro fichero escribimos:

```
/* main.css */
p {
  color: orange;
  font-size: 24px;
}
p {
  color: green;
}
```

En este caso el color de todas las etiquetas `<p>` será verde.

Ocurre lo mismo si la misma regla está definida en dos hojas CSS distintas que hayan sido cargadas usando la etiqueta `<link>`, en este caso prevalece el estilo definido en la última hoja cargada.

Si por error definiésemos la misma propiedad dos veces en un elemento también prevalecerá la segunda, por ejemplo:

```
/* main.css */
strong {
  background: orange;
  background: yellow;
}
```

En este caso el fondo de la etiqueta prevalecerá en amarillo.

4.3.2 Predominancia del estilo más específico

Ya hemos visto que podemos aplicar los estilos de tres formas, estas formas son de menos a más específicas:

- Por nombre de etiqueta
- Por clase (**class**)
- Por identificador (**id**)

Si asignamos estilos a un elemento de diferentes formas siempre predominará el más específico, esto quiere decir por ejemplo que si definimos lo siguiente:

```
<head>
  <style>
    #food {
      color: green;
    }
    p {
      color: orange;
    }
  </style>
</head>
<body>
  <p id="food">Mi fruta favorita es el mango.</p>
  <p>Mi cereal favorito es el trigo</p>
</body>
```



La frase "**Mi fruta favorita es el mango**" aparecerá en color verde, dado que un identificador es más específico que el nombre de la etiqueta. Sin embargo "**Mi cereal favorito es el trigo**" aparecerá en naranja, porque el estilo más específico para esa etiqueta es el de la etiqueta "**p**".

Nota: la especificidad se calcula de una manera más compleja como se puede ver en [este tutorial](#), pero para este curso he preferido simplificarlo un poco. Luego nos ayudaremos con DevTools para ayudar a experimentar y entender mejor cómo se aplican los niveles de especificidad.

4.3.3 Combinar selectores

Hasta el momento hemos visto cómo utilizar un selector para especificar un elemento, pero podemos combinar **cualquiera** estos selectores siguiendo las siguientes reglas:

- Si escribimos los selectores separados por un espacio estamos haciendo referencia a las etiquetas anidadas dentro de otras.
- Si añadimos los selectores sin separar por un espacio estamos haciendo referencia a un mismo elemento de una manera más específica.
- Para entender cómo se aplican los selectores debes leerlos de derecha a izquierda.
- El último selector antes del carácter "{" será al que se le aplique el estilo.

De este modo podemos seleccionar todos los elementos HTML que contienen una clase predefinida, etc. En este ejemplo vamos a ver cómo combinar nombres de etiquetas HTML con clases:

```
<head>
  <style>
    h1 span{
      color: red;
    }
    p.big{
      font-size: 2rem;
    }
    p.big span{
      font-weight: bold;
    }
  </style>
</head>
<body>
  <h1>Mi <span>Curriculum Vitae</span></h1>
  <p class="big"><span>Nombre</span>: Raúl Jiménez Ortega</p>
  <p class="big"><span>Lugar de nacimiento</span>: Málaga</p>
  <p>Fecha de nacimiento: 11/03/1984</p>
</body>
```

Aquí estamos indicando:

- Que el color del texto de las etiquetas **span** que sean descendientes (no necesariamente hijos directos) de las etiquetas **h1** aparezcan en rojo.
- Que los párrafos con que contengan la clase "**big**" tengan un tamaño de fuente de "**2rem**".
- Que el ancho de la fuente de las etiquetas **span** que sean descendientes de las etiquetas **p** con la clase **big** sea negrita.

Esto al principio puede parecer un poco lioso, pero al final verás que aprenderás a usarlo por sentido común.

Notas:

1. Del mismo modo podríamos hacer combinaciones usando identificadores, aunque dado que un identificador es único para un elemento y es el más específico no debería ser necesario usarlo nunca.
2. Al combinar los selectores la especificidad cambia como podemos leer en el tutorial antes mencionado.

4.3.4 Múltiples clases

Es habitual utilizar varias clases en un mismo elemento¹, por ejemplo:

```
<a class="btn btn-primary">Entrar</a>
```

A este elemento se le aplicarán los estilos de la clase ".btn" y la clase ".btn-primary".

Si se diese caso de que ambas clases especifican una misma propiedad, por ejemplo **color** predominará la de la última indicada, en este caso la de la clase ".btn-primary".

4.3.5 Otros selectores

Por último añadir que se puede aplicar el mismo estilo a varios selectores o conjunto de selectores separándolos por una coma, por ejemplo

```
.bold,  
strong,  
p.title{  
    font-weight: bold;  
}
```

Esto significa que tanto a las etiquetas con la clase "bold" como las "" como las "<p>" que contengan la clase "title" se les aplicará el estilo "font-weight: bold;".

También se puede usar el caracter ">" para especificar un hijo directo de un elemento, y otros pseudo-elementos, pero no entraremos en estos detalles en este curso.

Recursos CSS

- CSS3 Generator: <http://css3generator.com/>
- CSS3 Gradient Generator: <http://gradients.glrzad.com/>
- {CSS} Portal - on line generator: <http://www.cssportal.com/css3-rounded-corner/>
- CSS3 MAKER: <http://www.css3maker.com>
- CSS3.com: <http://www.css3.com/>
- **Kseso CSS:** CSS básico: Cascada, especificidad y herencia
<http://ksesocss.blogspot.com/2012/05/css-basico-cascada-especificidad-y.html>
- **Dev.Opera:** Recorriendo el árbol DOM
<http://dev.opera.com/articles/view/traversing-the-dom-es/>
- **LIBROS WEB:** Árbol de nodo
http://librosweb.es/javascript/capitulo_5/arbol_de_nodos.html
- **UOC:** Herencia y cascada
<http://mosaic.uoc.edu/ac/le/es/m6/ud2/>
- **LIBROSWEB:** Colisiones de estilos
http://librosweb.es/css/capitulo_2/colisiones_de_estilos.html
- **CSS:** Specificity Wars
http://www.stuffandnonsense.co.uk/archives/css_specificity_wars.html

MOOC CSS3

http://pigstroy.com/mooc-introduccion-al-desarrollo-web-parte-2-27-css-colores-css3-activate_5f7a0f6d1.html

4.4. SELECTORES BÁSICOS



Los selectores permiten seleccionar y manipular elementos HTML.

Estos son usados para encontrar o seleccionar elementos HTML basados en su id, clase, tipo, atributo y mas.

Los selectores identifican a un elemento dentro de la página Web para luego poder definir sus propiedades. Los distintos tipos de selectores abarcan desde el simple nombre de las etiquetas usadas en HTML (BODY, P, CODE, TABLE, UL, etc.) hasta complejas combinaciones que permiten un juego muy amplio de selecciones dentro de la página.

El conocimiento en profundidad de los distintos selectores es uno de los aspectos más complejos del lenguaje de las Hojas de Estilo y también el que nos permite sacar el máximo provecho de las CSS.

Existen muchos editores de páginas Web que facilitan la labor de crear y aplicar Hojas de Estilo, pero los selectores que se pueden definir con estos programas son sólo los más elementales. La complejidad que pueden alcanzar los selectores contextuales escapan a las posibilidades de cualquier herramienta y es ahí donde los diseñadores deberán apelar a sus propios recursos.

La estructura general de los selectores y atributos es de la siguiente manera:



4.4.1. 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.

4.4.2. 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; }
```

4.4.3. 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 `` 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 `` 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 `` contenidos dentro de un `<h1>`:

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

Con las reglas CSS anteriores:

- Los elementos `` que se encuentran dentro de un elemento `<p>` se muestran de color rojo.
- Los elementos `` que se encuentran dentro de un elemento `<h1>` se muestran de color azul.
- El resto de elementos `` 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 `` que se encuentren dentro de elementos de tipo ``, 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`.

4.4.4. 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>
```

¿Cómo se pueden aplicar estilos CSS sólo al primer párrafo? El selector universal (`*`) no se puede utilizar porque selecciona todos los elementos de la página. El selector de tipo o etiqueta (`p`) tampoco se puede utilizar porque seleccionaría todos los párrafos. Por último, el selector descendente (`body p`) tampoco se puede utilizar porque todos los párrafos se encuentran en el mismo sitio.

Una de las soluciones más sencillas para aplicar estilos a un solo elemento de la página consiste en utilizar el atributo `class` de HTML sobre ese elemento para indicar directamente la regla CSS que se le debe aplicar:

```
<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>
```

A continuación, se crea en el archivo CSS una nueva regla llamada `destacado` con todos los estilos que se van a aplicar al elemento. Para que el navegador no confunda este selector con los otros tipos de selectores, se prefija el valor del atributo `class` con un punto (`.`) tal y como muestra el siguiente ejemplo:

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

El selector `.destacado` se interpreta como *"cualquier elemento de la página cuyo atributo `class` sea igual a `destacado`"*, por lo que solamente el primer párrafo cumple esa condición.

Este tipo de selectores se llaman selectores de clase y son los más utilizados junto con los selectores de ID que se verán a continuación. La principal característica de este selector es que en una misma página HTML varios elementos diferentes pueden utilizar el mismo valor en el atributo `class`:

```
<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>
```

Los selectores de clase son imprescindibles para diseñar páginas web complejas, ya que permiten disponer de una precisión total al seleccionar los elementos. Además, estos selectores permiten reutilizar los mismos estilos para varios elementos diferentes.

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 `` 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`"*.

4.4.5. 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 { ... }
```

4.4.6. 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 { ... }
```

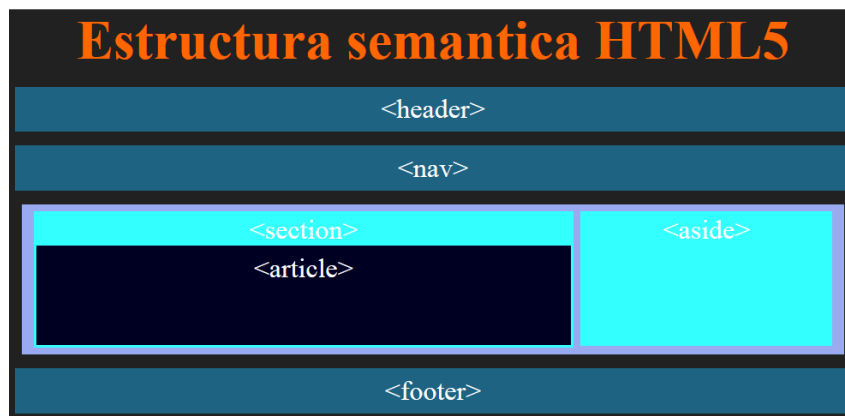
Ahora, el selector solamente selecciona aquellos elementos de tipo con un atributo class="especial" que estén dentro de cualquier elemento de tipo <div> que tenga un atributo class="aviso".

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

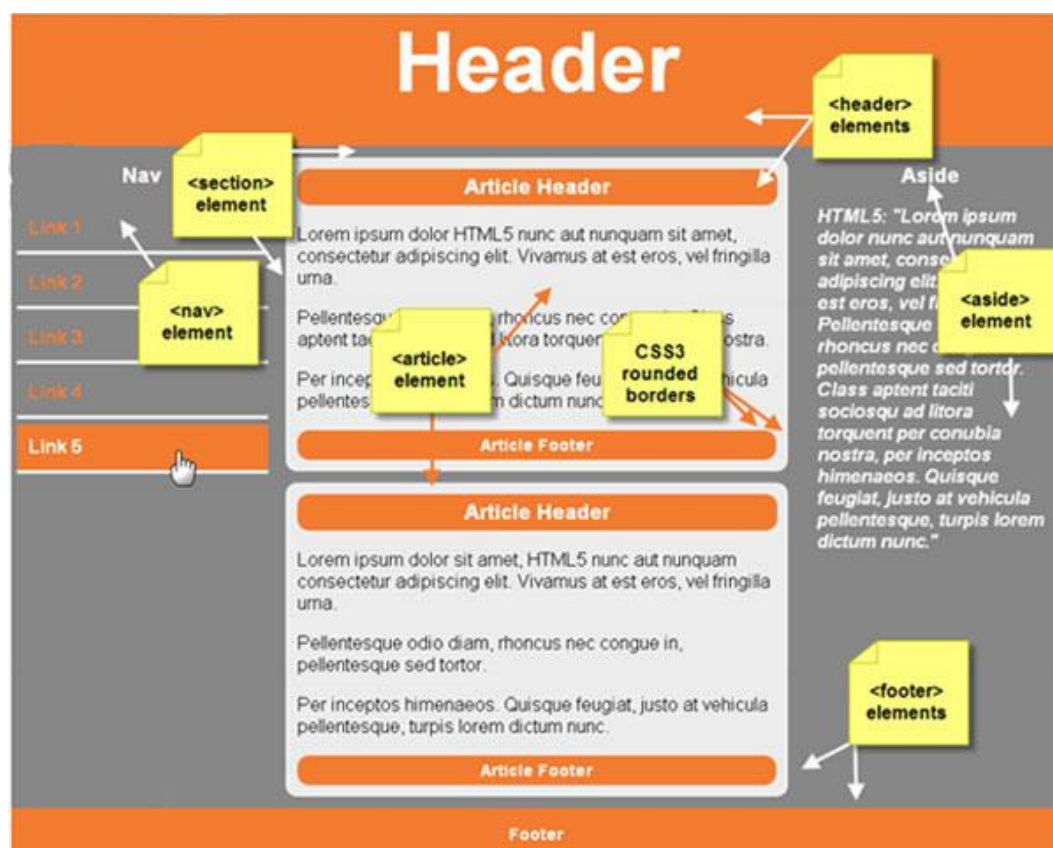
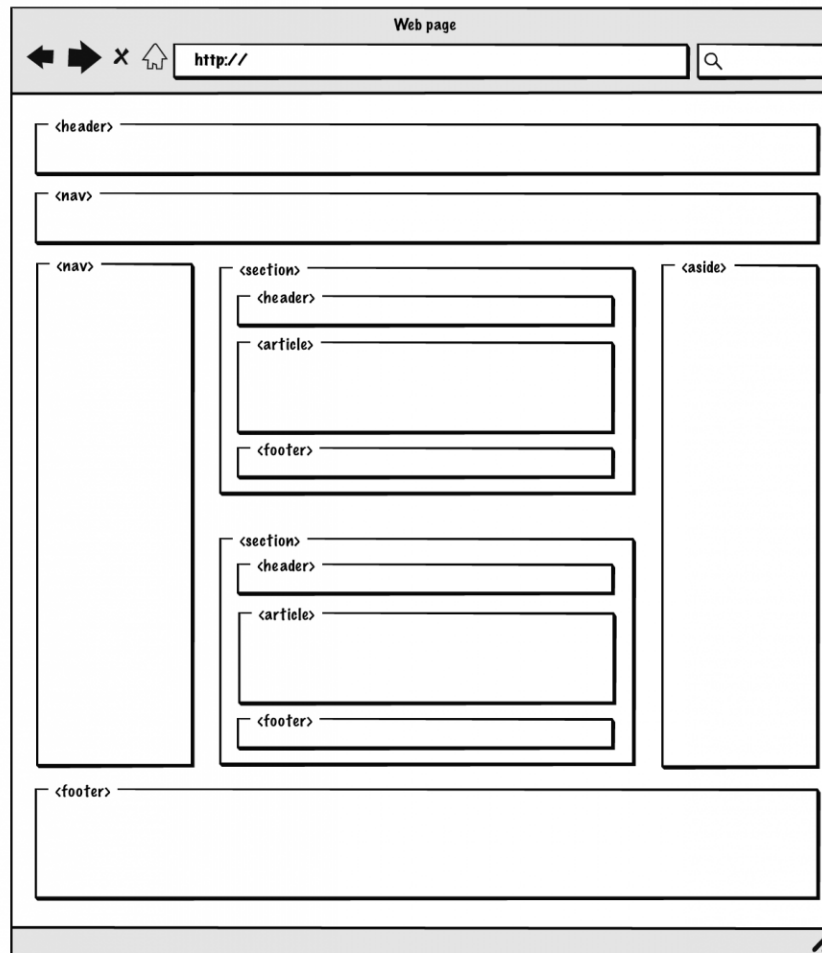
```
ul#menuPrincipal li.destacado a#inicio { ... }
```

El anterior selector hace referencia al enlace con un atributo id igual a inicio que se encuentra dentro de un elemento de tipo con un atributo class igual a destacado, que forma parte de una lista con un atributo id igual a menuPrincipal.

Ejemplo de maquetación en HTML con CSS



HTML	ESTILO
<pre> <!DOCTYPE html> <html lang="es"> <head> <title> Estructura Semantica en HTML5 y CSS3</title> <meta charset="utf-8"/> <link rel="stylesheet" type="text/css" href="estilo.css" /> <body> <h1> Estructura semantica HTML5</h1> <header> <header> </header> <nav> <nav> </nav> <section id="contenedor"> <section id="principal"> <section> <article> <article> </article> </section> <aside> <aside> </aside> </section> <footer> <footer> </footer> </body> </html> </pre>	<pre> * {margin: 0; padding: 0; } body{ background: #212121; /* #212121 escala de negro */ color:#FFF; /* #FFF color blanco */ font-size: 2em; } H1{ color: #F60; text-align:center; } header, nav, footer{ background: #1E6381; margin: 0.5em auto; max-width: 960px; padding:0.25em; text-align: center; } section#contenedor { background: #99aaf1; margin: 0.5em auto; max-width: 960px; padding: 0; text-align:center; } section#principal, aside{ background: #33ffff; display: inline-block; /* se comporta en bloque */ margin: 0.25em auto; max-width: 960px; min-height: 150px; padding: 0.1em; text-align: center; vertical-align: top; width: 65%; } article{ background: #000022; min-height: 100px; padding: 0.25em; } aside{ width: 30%; } </pre>



Autoevaluación

1. ¿Dónde introducimos los estilos (<style>) internos dentro del HTML?

- <head>
- <body>
- <header>

2. ¿Cuál es el código correcto?

- <span:color:red>
-
-

3. ¿Qué etiqueta es la correcta para enlazar el código CSS interno en el html?

- <link rel="stylesheet" type="css" url="style.css"/>
- <link rel="stylesheet" type="text/css" href="style.css"/>
- <style type="text/css">

4. Para maquetar una única palabra dentro de un texto para 'marcar' esta palabra utilizamos...

-
- <div>
- <type="text/css">

5. Para aplicar un mismo código CSS a múltiples html...

- ..utilizamos CSS interno
- ..utilizamos un fichero CSS externo..
- copiamos y pegamos el código CSS en cada html

6. ¿Los valores de estilos CSS por defecto del navegador, son iguales en todos los navegadores?

- Sí
- No

7. ¿Qué es más aconsejable, que el estilo esté en la cabecera del documento o en archivo CSS aparte?

- En la cabecera
- En hoja de estilo CSS aparte

8. ¿Qué es más aconsejable, que el estilo esté en la línea o en la cabecera del documento?

- En la línea
- En la cabecera