Referencias

Almacenar todos nuestros estilos en un archivo externo e insertar este archivo dentro de cada documento que lo necesite es muy conveniente, sin embargo no podremos hacerlo sin buenos mecanismos que nos ayuden a establecer una específica relación entre estos estilos y los elementos del documento que van a ser afectados.

Cuando hablábamos sobre cómo incluir estilos en el documento, mostramos una de las técnicas utilizadas a menudo en CSS para referenciar elementos HTML.

Existen varios métodos para seleccionar cuáles elementos HTML serán afectados por las reglas CSS:

- referencia por la palabra clave del elemento
- referencia por el atributo id
- referencia por el atributo class

Referenciando con palabra clave

Al declarar las reglas CSS utilizando la palabra clave del elemento afectamos cada elemento de la misma clase en el documento. Por ejemplo, la siguiente regla cambiará los estilos de todos los elementos <p>:

```
p {
        font-size: 20px;
 }
```

Si utilizamos la palabra clave p al frente de la regla le estamos diciendo al navegador que esta regla debe ser aplicada a cada elemento <p> encontrado en el documento HTML. Todos los textos envueltos en etiquetas <p> tendrán el tamaño de 20 pixeles.

Por supuesto, lo mismo funcionará para cualquier otro elemento HTML. Si especificamos la palabra clave span en lugar de p, por ejemplo, cada texto entre etiquetas tendrá un tamaño de 20 pixeles:

```
span {
        font-size: 20px;
 }
```

¿Pero qué ocurre si solo necesitamos referenciar una etiqueta específica? ¿Debemos usar nuevamente el atributo style dentro de esta etiqueta? La respuesta es no. Como aprendimos anteriormente, el método de Estilos en Línea (usando el atributo style dentro de etiquetas HTML) es una técnica en desuso y debería ser evitada. Para seleccionar un elemento HTML específico desde las reglas de nuestro archivo CSS, podemos usar dos atributos diferentes: id y class. Referenciando con el atributo id El atributo id es como un nombre que identifica al elemento. Esto significa que el valor de este atributo no puede ser duplicado. Este nombre debe ser único en todo el documento.

Para referenciar un elemento en particular usando el atributo id desde nuestro archivo CSS la regla debe ser declarada con el símbolo # al frente del valor que usamos para identificar el elemento:

```
#texto1 {

        font-size: 20px ;

}
```

La regla anterior será aplicada al elemento HTML identificado con el atributo id=”texto1”. Ahora nuestro código HTML quedará de esta manera:

```
<!DOCTYPE html>

<html lang="es">

<head>

        <title>Este texto es el título del documento</title>

        <link rel="stylesheet" href="estilos/estilos.css">

</head>

<body>

        <p id=”texto1”>Mi texto</p>

</body>

</html>
```

El resultado de este procedimiento es que cada vez que hacemos una referencia usando el identificador texto1 en nuestro archivo CSS, el elemento con ese valor de id será modificado, pero el resto de los elementos <p>, o cualquier otro elemento en el mismo documento, no serán afectados.

Esta es una forma extremadamente específica de referenciar un elemento y es normalmente utilizada para elementos más generales, como etiquetas estructurales. El atributo id y su especificidad es de hecho más apropiado para referencias en Javascript.

Referenciando con el atributo class

La mayoría del tiempo, en lugar de utilizar el atributo id para propósitos de estilos es mejor utilizar class. Este atributo es más flexible y puede ser asignado a cada elemento HTML en el documento que comparte un diseño similar:

```
.texto1 {

        font-size: 20px;

}
```

Para trabajar con el atributo class, debemos declarar la regla CSS con un punto antes del nombre. La ventaja de este método es que insertar el atributo class con el valor texto1 será suficiente para asignar estos estilos a cualquier elemento que queramos:

```
<!DOCTYPE html>
<html lang="es">
<head>
        <title>Este texto es el título del documento</title>
        <link rel="stylesheet" href="estilos/estilos.css">
</head>
<body>
        <p class=”texto1”>Mi texto</p>
        <p class=”texto1”>Mi texto</p>
        <p>Mi texto</p>
</body>
</html>
```

Los elementos <p> en las primeras dos líneas dentro del cuerpo del código en el ejemplo anterior tienen el atributo class con el valor texto1. Como dijimos previamente, la misma regla puede ser aplicada a diferentes elementos en el mismo documento. Por lo tanto, estos dos primeros elementos comparten la misma regla y ambos serán afectados por el estilo. El último elemento <p> conserva los estilos por defecto otorgados por el navegador.

La razón por la que debemos utilizar un punto delante del nombre de la regla es que es posible construir referencias más complejas. Por ejemplo, se puede utilizar el mismo valor para el atributo class en diferentes elementos pero asignar diferentes estilos para cada tipo:

```
p.texto1 {
        font-size: 20px;
 }
```

En este estilo estamos creando una regla que referencia la clase llamada texto1 pero solo para los elementos de tipo <p>. Si cualquier otro elemento tiene el mismo valor en su atributo class no será modificado por esta regla en particular.

Referenciando con cualquier atributo

Aunque los métodos de referencia estudiados anteriormente cubren un variado espectro de situaciones, a veces no son suficientes para encontrar el elemento exacto. La última versión de CSS ha incorporado nuevas formas de referenciar elementos HTML. Uno de ellas es el Selector de Atributo. Ahora podemos referenciar un elemento no solo por los atributos id y class sino también a través de cualquier otro atributo:

```
p[name] {
        font-size: 20px;
 }
```

En esta regla de estilo cambia solo elementos <p> que tienen un atributo llamado name. Para imitar lo que hicimos previamente con los atributos id y class, podemos también especificar el valor del atributo:

```
p[name=”mitexto”] {
        font-size: 20px;
 }
```

Hacemos referencia a los elementos <p> que tienen un atributo name con el valor mitexto. CSS3 permite combinar “=” con otros para hacer una selección más específica:

```
p[name^=”mi”] {
        font-size: 20px;
}
```

```
p[name$=”mi”] {
        font-size: 20px;
 }
```

```
p[name*=”mi”] {
        font-size: 20px;
 }
```

En CSS3 estos selectores producen los siguientes resultados:

• La regla con el selector ^= será asignada a todo elemento <p> que contiene un atributo name con un valor comenzado en “mi” (por ejemplo, “mitexto”, “micasa”).

• La regla con el selector $= será asignada a todo elemento <p> que contiene un atributo name con un valor finalizado en “mi” (por ejemplo “textomi”, “casami”).

• La regla con el selector *= será asignada a todo elemento <p> que contiene un atributo name con un valor que incluye el texto “mi” (en este caso, el texto podría también encontrarse en el medio, como en “textomicasa”).

En estos ejemplos usamos el elemento <p>, el atributo name, y una cadena de texto al azar como “mi”, pero la misma técnica puede ser utilizada con cualquier atributo y valor que necesitemos. Solo tiene que escribir los corchetes e insertar entre ellos el nombre del atributo y el valor que necesita para referenciar el elemento HTML correcto.

Referenciando con pseudo clases

CSS3 también incorpora nuevas pseudo clases que hacen la selección aún más específica.

```
<!DOCTYPE html>
<html lang="es">
<head>
	<title>Este texto es el título del documento</title>
	<link rel="stylesheet" href="estilos/estilos.css">
</head>
<body>
	<div id="wrapper">
		<p class="mitexto1">Mi texto1</p>
		<p class="mitexto2">Mi texto2</p>
		<p class="mitexto3">Mi texto3</p>
		<p class="mitexto4">Mi texto4</p>
	</div>
</body>
</html>
```

Miremos por un momento el nuevo código HTML anterior. Contiene cuatro elementos <p> que, considerando la estructura HTML, son hermanos entre sí e hijos del mismo elemento .

Usando pseudo clases podemos aprovechar esta organización y referenciar un elemento específico sin importar cuánto conocemos sobre sus atributos y el valor de los mismos:

```
p:nth-child(2){
	background: #999999;
}
```

Pseudo clase nth-child():

La pseudo clase es agregada usando dos puntos luego de la referencia y antes del su nombre. En la regla del estilo con párrafos referenciamos solo elementos <p>. Esta regla puede incluir otras referencias. Por ejemplo, podríamos escribirla como .miclase:nth- child(2) para referenciar todo elemento que es hijo de otro elemento y tiene el valor de su atributo class igual a miclase. La pseudo clase puede ser aplicada a cualquier tipo de referencia estudiada previamente.

La pseudo clase nth-child() nos permite encontrar un hijo específico. Como ya explicamos, el documento HTML del ejemplo tiene cuatro elementos <p> que son hermanos. Esto significa que todos ellos tienen el mismo padre que es el elemento .

Lo que esta pseudo clase está realmente indicando es algo como: “el hijo en la posición…” por lo que el número entre paréntesis será el número de la posición del hijo, o índice. Este estilo está referenciando cada segundo elemento <p> encontrado en el documento.

Usando este método de referencia podemos, por supuesto, seleccionar cualquier hijo que necesitemos cambiando el número de índice. Por ejemplo, la siguiente regla tendrá impacto sólo sobre el último elemento <p> de nuestra plantilla:

```
p:nth-child(4){

	background: #999999;

}
```

Como seguramente se habrá dado cuenta, es posible asignar estilos a todos los elementos creando una regla para cada uno de ellos:

```
*{

	margin: 0px;

}

p:nth-child(1){

	background: #999999;

}

p:nth-child(2){

	background: #CCCCCC;

}

p:nth-child(3){

	background: #999999;

}

p:nth-child(4){

	background: #CCCCCC;

}
```

En la primera regla usamos el selector universal * para asignar el mismo estilo a cada elemento del documento. Este nuevo selector representa cada uno de los elementos en el cuerpo del documento y es útil cuando necesitamos establecer ciertas reglas básicas. En este caso, configuramos el margen

de todos los elementos en 0 pixeles para evitar espacios en blanco o líneas vacías como las creadas por el elemento <p> por defecto.

En el resto del código del ejemplo usamos la pseudo clase nth-child() para generar un menú o lista de opciones que son diferenciadas claramente en la pantalla por el color de fondo.

Para agregar más opciones al menú, podemos incorporar nuevos elementos <p> en el código HTML y nuevas reglas con la pseudo clase nth-child() usando el número de índice adecuado. Sin embargo, esta aproximación genera mucho código y resulta imposible de aplicar en sitios webs con contenido dinámico. Una alternativa para obtener el mismo resultado es aprovechar las palabras clave odd y even disponibles para esta pseudo clase:

```
*{
	margin: 0px;
}
p:nth-child(odd){
	background: #999999;
}
p:nth-child(even){
	background: #CCCCCC;
}
```

Ahora solo necesitamos dos reglas para crear la lista completa. Incluso si más adelante agregamos otras opciones, los estilos serán asignados automáticamente a cada una de ellas de acuerdo a su posición. La palabra clave odd para la pseudo clase nth-child() afecta los elementos <p> que son hijos de otro elemento y tienen un índice impar. La palabra clave even, por otro lado, afecta a aquellos que tienen un índice par. Existen otras importantes pseudo clases relacionadas con esta última, como first- child, last-child y only-child, algunas de ellas recientemente incorporadas. La pseudo clase first-child referencia solo el primer hijo, last-child referencia solo el último hijo, y only-child afecta un elemento siempre y cuando sea el único hijo disponible. Estas pseudo clases en particular no requieren palabras clave o parámetros, y son implementadas como en el siguiente ejemplo:

```
*{
	margin: 0px;
}
p:last-child{
	background: #999999;
}
```

Otra importante pseudo clase llamada not() es utilizada realizar una negación:

```
:not(p){
        margin: 0px;
}
```

La regla anterior asignará un margen de 0 pixeles a cada elemento del documento excepto los elementos <p>. A diferencia del selector universal utilizado previamente, la pseudo clase not() nos permite declarar una excepción. Los estilos en la regla creada con esta pseudo clase serán asignados a todo elemento excepto aquellos incluidos en la referencia entre paréntesis. En lugar de la palabra clave de un elemento podemos usar cualquier otra referencia que deseemos. En el próximo listado, por ejemplo, todos los elementos serán afectados excepto aquellos con el valor mitexto2 en el atributo class:

```
:not(.mitexto2){
        margin: 0px;
}
```

Cuando aplicamos la última regla al código HTML del ejemplo anterior el navegador asigna los estilos por defecto al elemento <p> identificado con el atributo class y el valor mitexto2 y provee un margen de 0 pixeles al resto.

Nuevos selectores

Hay algunos selectores más que fueron agregados o que ahora son considerados parte de CSS3 y pueden ser útiles para nuestros diseños. Estos selectores usan los símbolos >, + y ~ para especificar la relación entre elementos.

```
div > p.mitexto2{
        color: #990000;
}
```

Selector >

El selector > está indicando que el elemento a ser afectado por la regla es el elemento de la derecha cuando tiene al de la izquierda como su padre. La regla anterior modifica los elementos <p> que son hijos de un elemento . En este caso, fuimos bien específicos y referenciamos solamente el elemento <p> con el valor mitexto2 en su atributo class.

El próximo ejemplo construye un selector utilizando el símbolo +. Este selector referencia al elemento de la derecha cuando es inmediatamente precedido por el de la izquierda. Ambos elementos deben compartir el mismo padre:

```
p.mitexto2 + p{
        color: #990000;
}
```

Selector +

La regla anterior afecta al elemento <p> que se encuentra ubicado luego de otro elemento <p> identificado con el valor mitexto2 en su atributo class. Si abre en su navegador el archivo HTML con el código del ejemplo, el texto en el tercer elemento <p> aparecerá en la pantalla en color rojo debido a que este elemento <p> en particular está posicionado inmediatamente después del elemento <p> identificado con el valor mitexto2 en su atributo class.

El último selector que estudiaremos es el construido con el símbolo ~. Este selector es similar al anterior pero el elemento afectado no necesita estar precediendo de inmediato al elemento de la izquierda. Además, más de un elemento puede ser afectado:

```
p.mitexto2 ~ p{

        color: #990000;

}
```

Selector ~

La regla anterior afecta al tercer y cuarto elemento <p> de nuestra plantilla de ejemplo. El estilo será aplicado a todos los elementos <p> que son hermanos y se encuentran luego del elemento <p> identificado con el valor mitexto2 en su atributo class. No importa si otros elementos se encuentran intercalados, los elementos <p> en la tercera y cuarta posición aún serán afectados. Puede verificar esto último insertando un elemento mitexto luego del elemento <p> que tiene el valor mitexto2 en su atributo class. A pesar de este cambio solo los elementos <p> serán modificados por esta regla.