Universidad Autónoma Metropolitana,

2020

Sergio Zepeda Hernández

Pre-borrador de libro.

Tip 28 Identificación grupal

En este momento ya podemos darle estilo a cualquiera de los elementos en un documento html, Pero existe una pequeña problemática cuando nosotros en la hoja de estilo especificamos que un elemento como tenga un cierto estilo, esto afectará a todos los elementos span que estén incluidos en el documento html. Es decir, todas estas etiquetas son afectadas con ese estilo predefinido. En algunas ocasiones esto puede ser suficiente en un diseño, pero en otros más elaborados se requiere que la misma etiqueta tenga diferentes estilos dependiendo de quien la contiene, posición, lugar o importancia y que otras etiquetas compartan estilos, en otras palabras etiquetas que compartan un mismo estilo.

Para ello, nosotros podemos hacer uso de lo que denominamos una “clase” la cual permite definir un estilo específico y el cual puede ser ocupado por cualquier otra etiqueta html si esta es incluida como parte de ella. Es decir, yo generó una clase con un estilo específico, nombró a esa clase y la incluyó como parte de las propiedades de la etiqueta html y el estilo predefinido es aplicado en la etiqueta.

css

```
span{
color:orange;
}
```

```
.colorazul{
background-color: Deepskyblue;
color:blue;
}
```

Podemos observar que si definimos un estilo para span este afectará a todos los que están incluidos en el documento. pero en el código de abajo tenemos una “clase” definida como “.colorazul”, es importante notar que empieza con un punto, esto hace saber al navegador que es un clase que será ocupada por varias etiquetas.

Para especificar en el documento html que la clase color azul afecte a una etiqueta en específico, es necesario colocar en la etiqueta de apertura y en sus propiedades el nombre de esta con la palabra reservada class de la siguiente manera: “class=”*nombreclase*”. A continuación vamos a ver un ejemplo de su uso:

html

```
<span> texto con estilo global o sin estilo </span>
<span class=”colorazul” >  mi texto con estilo</span>
<span class=”colorazul” > otro texto con estilo </span>
<p class=”colorazul” > todo mi texto ……..  </p>
```

Ya en el código html podemos ver que la primera etiqueta tendrá el primer estilo donde se espifica el color de la letra en naranja, pero las siguientes etiquetas donde ya se especifica la clase “colorazul” (**aquí sin punto) el estilo en esa clase afectarán a los dos span siguientes, a su vez, esta misma clase puede ser ocupada por otras etiquetas, en este caso <p> también ocupa la clase y el estilo la afectara.
Esta es una gran ventaja porque nos permite predecir estilos que podemos ocupar continuamente, es decir, un reutilización continua de código ya escrito, esto es especialmente útil cuando necesitamos mantener un estilo para diferentes documentos html que pertenecen a un mismo sistema.
**Actividad
Crear un documento con 10 etiquetas html diferentes o iguales y cree una clase para modificar 5 de ellas.

Tip 29 Identificación única
Una vez entendido cómo generar estilos que pueden ser reutilizables varias veces, también es necesario conocer cuando deseamos estilizar de manera muy específica a una sola etiqueta html en particular y generarle un estilo muy específico. La manera de hacerlo es crear un identificador

único y éste es especificado en el archivo css como un símbolo " # "y en el documento html a través de un "Id" veamos el siguiente ejemplo

CSS

```
#parrafoverde{
background-color:Turquoise;
color:Teal;
}
```

Html

```
<p id="parrafoverde" > texto ……. </p>
```

Figura R

La figuraR muestra cómo se declara el estilo en el archivo css A través del símbolo #seguido del nombre del identificador. Por otra parte, en el documento html la etiqueta debe incluir al siguiente sintaxis : id="nombre_unico" , en este ejemplo es: "*parrafoverde*". Es muy importante resaltar que **no puede haber otro elemento con un Id igual,** ya que éste debe de ser único en todo el documento html.

*** actividad
Incluya 7 etiquetas html y genere para tres de ellas un id con un respectivo estilo.

Tip 30 comentarios en una hoja de estilo

De aquí en adelante vamos a ingresar comentarios en nuestro archivo de estilo CSS, la manera de hacerlo es a través de los siguientes símbolos de inicio " /* " y de final " */" todo lo que este contenido al interior de estos símbolos no será tomado en cuenta por el navegador, hay que recordar que los comentarios sirven para dar significado al código que vamos escribiendo, y sirva de referencia en un futuro para modificaciones o mantenimiento. Siempre es importante antes de liberar el código tener un archivo con estos comentarios.

```
/*  esto es un comentario   */
```

***Actividad
Ponga comentarios a un archivo css.

*****De ahora en adelante ponga comentarios a su código CSS para poder ser revisado por el profesor a cargo, si no hay comentarios no se revisarán sus actividades****

Tip 31 capas

Ahora Necesitamos aprender de una de las etiquetas más importantes en un documento html la etiqueta denominada como , la cual sirve para contener uno o múltiples etiquetas html. Esta actúa como un contenedor y un separador de código. A su vez, nos permite estructurar de manera controlada gran parte de un documento html. Cuando nosotros logramos estilizar de una manera particular un nos permite generar interfaces muy agradables para la interacción con los usuarios. Un dív puede ser definido como una sección, una división, un contenedor, y muchos lo refieren como una capa, esto por la similitud que tiene cuando nosotros utilizamos un editor de imágenes que funciona a través de diferentes capas, las cuales pueden unirse, separar elementos o sobreponerse para generar un diseños muy estéticos, en nuestro caso en específico, nosotros las denominaremos como capas. Y así nos referiremos a este etiqueta de aquí en adelante.

En términos prácticos un actúa como un minibody dentro del body, es decir, éste puede contener diferentes elementos o etiquetas html en su interior, pero se puede utilizar de una manera muy particular permitiendo generar estética en el documento y una estructura y lógica coherente del documento y la información a visualizar a los usuarios.

```
<div>
        <span>  texto    </span>
         <p>   paraffo   </p>
        <a href= " " >   link   </a>
</div>
```

***Actividad

Cree un nuevo documento html que incluya mucha información en títulos y párrafos y algunas imágenes y utilice al menos 4 capas para contener esos elementos. es muy importante mantener una identación al interior de cada capa y debe estar presente de aquí en adelante, ya que podemos tener capas anidadas, sobrepuestas o una al lado de otra, etc. Y con la identación podemos localizar de manera muy rápida código a modificar o mantener.

Tip 32 fondo en una capa

Estilizar una capa es una de las partes más importantes que debe de conocer un desarrollador en html5, debido a que si se tiene un pobre conocimiento en este sentido, sólo se generan interfaces difíciles de usar con una estética muy deficiente. La manera más sencilla de estilizar una capa en un archivo css es por medio dl código que afectará a todas las capas dentro de body:

```
div{
/* código de estilo  */
}
```

Figura S

Si lo hacemos de la manera en que se muestra en la figura S, podemos ver que todos los divs serán afectados con ese único estilo, para este tip vamos a generar un Id por cada capa y probemos las siguientes instrucciones para conocer cada una de ellas:

background-color: *color*;
Ya conocemos esta instrucción y sirve para cambiar el color, de aquí en adelante se recomienda al estilizar la posición, tamaño, etc. dar un color provisional para poder visualizar las instrucciones de estilo, una vez que se logra el efecto deseado esta propiedad se puede cambiar o eliminar. Esto es importante porque muchas veces hay visualizaciones de capas que no entendemos qué sucede, por que la capa no es visible. Así que es muy importante recordar en todo momento este consejo, y evitar múltiples dolores de cabeza o frustraciones.

background-image: src=url("../directorio/image.jpg ");

Sirve para incorporar una imagen como fondo, lo único que necesitamos es incluir esta etiqueta y la ruta del archivo entre paréntesis y comillas a través de los paréntesis dar la ruta del archivo de imagen que será incluido como fondo.

background-repeat: repeat-x; /* repeat-y, no repeat, */

Esta instrucción sirve básicamente cuando se tienen imágenes muy pequeñas que pueden producir efectos muy interesantes al ser repetidas ya sea en el eje de las x, como en el eje de las y, o si se desea especificar que no se desea repetir una imagen. Se muestran los valores en el comentario para poder experimentar con ellos.

background-attachment: fixed; /* scroll */

Instrucción para cuando se desea que una imagen permanezca de manera fija, aunque la barra de desplazamiento corra de un lado a otro mostrando todo el contenido.
background-position: center center;

/*left top, left center, left bottom, center, center center, center top, center bottom, right top, right center, right bottom */

Sirve para posicionar una imagen como fondo dentro de una etiqueta body o div, se pueden probar todas las opciones para ver el efecto visual que produce cada valor, de preferencia usar una imagen pequeña.

***Actividad
pruebe poner color e imágenes en el fondo de los cuatro divs de su archivo anterior y pruebe las instrucciones de este tip.

Tip 33 Tamaño de una capa

Para muchas interfaces de un sistema es muy importante predefinir el tamaño de una capa, ya que esta convivirá con algunas otras y si deseamos un comportamiento controlado debemos conocer las siguientes instrucciones.

```
/* definiendo alto y ancho en porcentaje o pixeles  según prefiera */
width: 50%;
height: 200px;

/* delimitar el tamaño  máximo */
max-width: 600px;
max-height: 30%;
/* delimitar el tamaño  minimo */
min-width: 500px
min-height: 100px;

/*pintar una línea al límite de la capa */
border: 1px solid red;
/*  márgenes de visualización  */
margin: auto;   /* margen automático */
 /* para cada lado  */
margin-top:100px;
margin-right:120px;
margin-bottom:120px
margin-left:30px
/* las cuatro instrucciones juntas */
margin: 25px 50px 75px 100px;
```

***Actividad

Pruebe las diferentes instrucciones, cambie valores y defina el tamaño de sus cuatro capas de su archivo generado en el tip anterior.

Tip 34 Posición capa

El posicionamiento de las capas es una de las partes más importantes en el diseño de interfaces html5, muchos programadores tienen poco conocimiento y eso ocasiona que se utilicen plataformas de desarrollo que dicen generar código estándar en realidad no lo es, ya que a la hora de visualizar el diseño en diferentes navegadores fracasa de un navegador a otro, por eso es muy recomendable e imperativo comprender el funcionamiento y significado de código estándar. Actualmente las hojas de estilo nos ofrecen una gran variedad de instrucciones que nos permiten muchas funcionalidades que antes sólo eran posibles con JavaScript . Ahora dominar las hojas de

estilo es una gran ventaja para obtener un muy buen diseño de interfaz y proporcionarle usabilidad a los usuarios. El posicionamiento de una capa en la mayoría de veces, puede parecer un proceso muy complejo y complicado y solo para expertos, en la realidad no lo es tanto si solo se obtiene una explicación clara y concisa sobre el significado de cada instrucción. A continuación la instrucción es la siguiente (recuerde tener un color o margen en la capa y un id para cada capa para dar las instrucciones correspondientes) :
CSS

```
position: static;     /* relative, fixed, absolute, sticky  */
```

La instrucción posición tiene cinco valore los cuales son descritos a continuación:
static : Este es el valor estándar y qué es asignado por default cuando la instrucción no es especificada, este valor indica que la capa no tiene una posición especial y sólo es visualizada conforme al flujo de la página.
relative: Este valor es ocupado Cuando deseamos que la posición en la capa se mueva a alguno de los cuatro lados, y se ocupa junto con left, right, top, bottom, para definir la posición con respecto a alguno de los lados de la pantalla. ejemplo:

```
position: relative;
right: 150px;
```

fixed**:** en términos muy sencillos este valor fija una capa a una posición específica (izquierda, derecha, arriba o abajo) y está quedará inmovible aún cuando la barra de desplazamiento mueva todo el contenido para probar este valor es importante tener muchos elementos en la página para poder deslizar la barra de desplazamiento y ver el efecto de fijación. Probemos las siguientes instrucciones

```
position: fixed;
right: 50px;          /* left  */
top:250px;          /* bottom */
width: 25%;
height: 30%;
```

sticky: este valor es ocupado básicamente para alcanzar un límite en la parte superior y fijar la capa en ese límite la instrucción necesita que la palabra reservada “-webkit” y hacerla compatible con algunas versiones de navegadores. Para ver el efecto es necesario tener contenido y la capa en la parte de abajo y al subir la barra de desplazamiento y la capa alcanzar los 30px en la parte superior ,esta será fijada.

```
position: -webkit-sticky;
position: sticky;
 top: 30px;
```

absolute **:** El valor absoluto nos sirve básicamente para posicionar una capa que está dentro de otra capa, es decir, la posición absoluta se ocupa para que capas hijas que están contenidas en una capa madre, así los valores de estilo son con respecto al interior de la capa principal, madre y contenedora, así para probar estas instrucciones genere una capa con posición y luego en su interior crear una nueva capa y dar el siguiente estilo, así podemos observar como los valores de bottom (abajo) , right (derecha), son con respecto al interior de la capa madre.

```
position: absolute;
bottom: 30px;
right: 20px;
width: 200px;
 height: 100px;
```

***** Actividad.**

***** Genera un archivo html y pruebe cada uno de los valores de posicionamiento de capas y comente el archivo CSS sobre las instrucciones de posición.**

Tip 35 Traslape de capas o imágenes

Cuando se ocupa la instrucción “posición” y las capas nos quedan traslapadas, podemos poner un orden a ese traslape por medio de la propiedad “z-index”, a la cual le podemos dar valores y ordenar su posición arriba o abajo como si fueran capas en un editor de imágenes u hojas de papel sobrepuestas, así entre mayor es el número más arriba está, entre menor sea el número inclusión números negativos, más abajo está la capa. En algunos diseños se puede necesitar este efecto y esta es la mejor manera de hacerlo cuando de trata de capas o imágenes. Un ejemplo de su uso como instrucciones de estilo es como sigue.

```
#capa5 {
  position: absolute;
   z-index: -1;
}
```

****Actividad

Genere 4 capas propias y utiliza las propiedades “position “ y “z-index” para cada una de ellas e intercambia la posición para mostrar el efecto de traslape.

Tip 36 capas flotantes

Una capa flotante es una de las propiedades menos entendidas y causa mucha confusión debido a que no se tiene un entendimiento claro de su función. Para entender el funcionamiento vamos a generar dos capas la primera con el id de “uno”, la cual contendrá un párrafo de texto de una 15 líneas aproximadamente. La segunda capa con id “dos” debe estar anidada dentro de la capa “uno” como en el ejemplo anterior y deberá contener un texto de 8 líneas aproximadamente “o”

una imagen con un ancho de 150 píxeles y de alto 350 píxeles. Ahora probemos el siguiente código CSS y veamos la visualización:

```
#uno {
        background-color:cyan;

}

#dos {
        float: right;
        background-color:blue;
        color:yellow;
        width: 150px;
        hight: 350px;
}
```

Si visualizamos el documento lo que podemos observar es que la capa “uno” es más pequeña que la capa “dos” en altura, pero la capa “dos” flota sobre el lado derecho y esta sale fuera del rango de la capa contenedora. En términos sencillos utilizar float, me permite que una capa o una imagen pueda flotar al lado de texto, rebasando incluso a su capa contenedora, porque es como si flotara arriba de ella.
Para dar solución a esta problemática, es necesario utilizar la siguiente instrucción “overflow:auto;” en el estilo de la capa “uno” de la siguiente manera:

```
#uno {
        background-color:cyan;
        overflow:auto;
}
```

Ahora se puede observar que la capa madre “uno” ya puede contener a la capa hija “dos” dentro de sus límites, entender este proceso nos ayudará de manera muy importante cuando desarrollemos una interfaz responsiva.

***Actividad**
***genere un archivo html que aplique lo aprendido en el tip 35 sobre capas flotantes, recuerde poner comentarios a su archivo css. Proporcionar más estilo a cada capa.

Tip 37 Controlar flotantes
Cuando se tienen capas separadas y se utiliza la propiedad de flotante en el estilo de alguna de ellas, se puede presentar la misma problemática de tener capas sobrepuestas una sobre la otra veamos el siguiente ejemplo, el código html sería el siguiente:

```
<div id="capa1">Contenido capa1</div>

<div id="capa2">Este es el contenido de la capa2 la cual tiene un párrafo de texto, Este es el contenido de la capa2 la cual tiene un párrafo de textoste es el contenido de la capa2 la cual tiene un párrafo de texto, Este es el contenido de la capa2 la cual tiene un párrafo de textoste es el contenido de la capa2 la cual tiene un párrafo de texto, Este es el contenido de la capa2 la cual tiene un párrafo de texto,Este es el contenido de la capa2 la cual tiene un párrafo de texto,Este es el contenido de la capa2 la cual tiene un párrafo de texto</div>
```

Podemos ver el código html dónde podemos observar dos capas separadas, ahora en las hojas de estilo vamos a escribir el siguiente código:

```
#capa1 {
        float: right;
        background-color:beige;
        border: 1px solid blue;
       width: 150px;
       height: 80px;
       margin: 20px;
       padding: 10px;
}

#capa2 {
       background-color:cyan;
        border: 1px solid green;
 }
```

Si visualizamos el contenido podemos observar que la capa 2 queda debajo de la capa 1 y estas quedan sobrepuestas o traslapadas. Aquí pareciera que el problema es mucho más complejo, pero en la realidad se soluciona de una forma muy sencilla, sólo necesitamos decirle a la capa 2 que elimine la propiedad de flotante sobre ella, al realizar esto las capas quedan nuevamente separadas e independientes entre sí, la manera de hacerlo es incluyendo la línea clear y limpiar el lado donde se encuentra la capa flotante que necesitamos liberar. Podemos liberar a la derecha, izquierda o ambos (both), ninguna (none) es la opción por default cuando la instrucción no está especificada.

```
#capa2 {
       background-color:cyan;
        border: 1px solid green;
       clear:right;            /* none, left, right, both */
 }
```

***Actividad

*** Hacer capas 1 y dos con información propias, darles color y propiedades de estilo y mostrar la aplicación de float y clear en ellas.

Tip 38 Transparencia

Una de las propiedades más importantes tiene que ver con la transparencia ya que está nos permite ofrecer unos diseños muy estéticos y novedosos. Una hoja de estilo nos permite dar transparencia al contenido de varias etiquetas html como texto, capas, imágenes, etc. Es importante recordar que si aplicamos la transparencia a una capa está afecta a todo su contenido, por lo que para establecer ciertos efectos es necesario aplicar esta propiedad a través de un "id".y evitar su uso en "class" o elementos que tienen mucho contenido. El rango de valores utilizados para esta propiedad es de 0 a 1 en donde 0 es totalmente transparente o invisible y 1 donde no se tiene nada de transparencia, y siendo 0.5 un valor con mediana transparencia. Se puede usar cualquier valor en este rango para dar el efecto deseado, un ejemplo de su de su uso en el estilo es:

opacity: 0.3;

***Actividad**

Genere una página con contenido propio y aplique en tres elementos la propiedad de opacity.