

TEMA 1: INTRODUCCIÓN A LAS HOJAS DE ESTILO.

Para establecer el formato de un documento en HTML clásico tiene limitaciones, por este motivo los autores de páginas Web han ido desarrollando una serie de trucos que a veces causan problemas, por otro lado en el HTML clásico esta mezclado en contenido del documento con las etiquetas necesarias para darle formato, esto dificulta la búsqueda de errores y la depuración de las páginas Web, para resolver estos problemas se introdujeron las hojas de estilo.

Las hojas de estilo permiten definir mediante una sintaxis especial el formato a distintos niveles:

- 1.- Un sitio Web completo.
- 2.- Un documento HTML o página Web.
- 3.- Una sección de un documento.
- 4.- Una etiqueta concreta.
- 5.- Pequeños textos o trozos de un documento.

Además se puede aplicar un formato mucho mas preciso, por ejemplo, se puede definir la distancia entre líneas, se pueden aplicar sangrías a voluntad, se pueden colocar los elementos con precisión en cualquier lugar de la página, se puede hacer que un elemento sea visible o no, se pueden establecer capas,...

Se pueden utilizar más unidades de medida (8pulgadas, punto o centímetros).

En resumen, las hojas de estilo permiten mejorar la apariencia o presentación de las páginas Web.

Cada navegador tiene sus criterios para presentar una Web, las hojas de estilo permiten redefinir las reglas que utiliza el navegador para presentar una página Web.

Hay distintos modelos de hojas de estilo, las más utilizadas son las hojas de estilo en cascada (CSS) se llaman así porque permiten que las informaciones de estilo provenientes de varias fuentes pueden combinarse, la cascada define una secuencia ordenada de hojas de estilo en la que las reglas de los estilos mas concretos tienen prioridad sobre los mas genéricos.

Nota: en principio HTML no esta ligado a ningún lenguaje de hojas de estilo concreto por lo que puede utilizar cualquier lenguaje de hijas de estilo. No obstante en la última especificación del HTML 4.01 se hace referencia a las CSS de nivel 1.

ESPECIFICACIONES DEL LENGUAJE DE HOJAS DE ESTILO POR DEFECTO.

Según la especificación HTML 4.01 del W3C al crear más páginas Web se debe especificar el lenguaje de hojas de estilo asociado con la página, para ello se utiliza una etiqueta meta dentro de la cabecera de la página, si el valor por defecto del lenguaje de estilo es CSS se usará la siguiente etiqueta:

```
<Meta http-equiv="content-stily-type", content="text/css">
```

Además en dicha especificación se recomienda que si no se especifica ningún lenguaje de estilos el navegador tome como lenguaje por defecto el CSS.

Para enlazar una página Web con el archivo de estilos se utiliza la etiqueta <link>, carece de etiqueta de cierre y se utilizan los siguientes atributos:

- rel="stylesheet"; indica un enlace a una hoja de estilos
- Type="text/css"; indica que se va a utilizar una sintaxis css.
- Href="nombre archivo que contiene los estilos.css"

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

Nota: una hoja de estilo en un archivo independiente o una declaración de estilo en la cabecera de una página demasiado compleja puede retrasar la presentación del documento al usuario. Por este motivo la especificación 4.01 de HTML permite los estilos a nivel de sección, a nivel de etiqueta y a nivel de trozos de texto.

AGRUPACIÓN DE ATRIBUTOS.

a) varias etiquetas tienen los mismos atributos, si hemos definido varias etiquetas con los mismos atributos se pueden agrupar escribiendo el nombre de las etiquetas separadas por comas y a continuación los atributos entre llaves, ejemplo:

```
H2,H3,H4 {  
    Color:#000ff;  
    Text-align:left;  
}
```

b) Agrupación de los atributos FONT; en vez de escribir el nombre de cada uno de ellos separados por comas, podemos escribir un solo FONT y sus valores

```
DIV {font-weight:bold;font-style:Italic;font-size:12pt;font-family:serif} → Se puede  
sustituir por:  
DIV {font:bold italic 12Pt serif}
```

c) Agrupación de los atributos de los márgenes:

```
body {margin-top:20px; margin-right:10px; margin-left:10px; margin-bottom:20px} →  
Se puede sustituir por:  
Body {margin:20px 10px 10px 20px}
```

ESTILO DE PEQUEÑOS TEXTOS.

Para darles el estilo a pequeños grupos de texto a un fragmento de texto se utiliza:

```
<Span style="color:green; font-weight:bold;">.....</Span>.
```

Todo lo que halla entre y permite cambiar el estilo a esa zona.

ESTILO PARA UNA ETIQUETA.

Se puede hacer que el contenido de una etiqueta completa muestre un estilo determinado, para ello se utiliza el atributo style dentro de la etiqueta:

```
<P style="atributo1:valor; atributo2:valor;">.
```

ESTILO DE UNA SECCIÓN DE UN DOCUMENTO.

Puede hacerse con el atributo Style dentro de una etiqueta DIV:
<Div Style="color:#0000ff; font-weight:bold;">

Si dentro de una sección establecemos una etiqueta o un estilo para un trozo de texto predomina sobre lo general.

SINTAXIS CSS.

1.- Para definir el estilo de un texto se utilizan atributos (fotocopia) seguido del carácter ; se pueden separar varios elementos y todos ellos se escribirán entre comillas, igualadas al atributo Style.

Cuando se definen los estilos para toda una página Web o para todo un sitio Web se utilizará la siguiente sintaxis:

2.- Para definir el estilo de una etiqueta se escribe la etiqueta seguida de los atributos entre llaves, ejemplo: H1 {text-decoration:underline; text-align:center;}.

3.- Los valores asignados pueden llevar una unidad de medida que puede ser:

Pt → Puntos.

In → Pulgadas.

Cm → Centímetros.

Px → Píxeles.

4.- Las etiquetas para las que no se defina un estilo se comportarán siguiendo las normas predefinidas para el navegador.

ESTILOS PARA UNA PÁGINA COMPLETA.

Se utiliza la etiqueta <Style> y </Style> dentro de la cabecera del documento, sus principales atributos son:

- Type; que se hace igual al lenguaje de hojas de estilo utilizado entre comillas <style type="text/css">. Este valor prevalece sobre el lenguaje de hojas de estilo por defecto.

- Media; será igual al medio en que se va a presentar la información, por defecto el valor es screen:

<Style media="screen">, dentro de media se pueden especificar varios medios separados por comas.

Se pueden utilizar tantas etiquetas <Style> </Style> dentro de la cabecera.

Se aplican estilos a la etiqueta body serán heredados por el resto de las etiquetas del documento, salvo que se definan estilos para ciertas etiquetas en cuyo caso el estilo definido predomina sobre el general.

Los estilos se engloban dentro de la etiqueta comentario en HTML `<!--.../-->`, esto se hace para que no sean visualizados por los navegadores que no soportan la sintaxis css.

COMENTARIOS EN HOJAS DE ESTILO.

Se pueden añadir comentarios a los estilos escribiéndolos entre `/*` y `*/`.

ESTILOS PAA UN SITIO WEB COMPLETO.

El estilo se establece en un archivo independiente en el que sólo existen definiciones de estilos de páginas y enlazando todas las páginas del sitio Web, las que se deseen, con el archivo, de este modo basta con cambiar el/los archivo/os de estilos para cambiar el estilo de varios o de todas las páginas.

El archivo de estilos es un archivo de texto que puede tener cualquier extensión, pero lo mejor es ponerle la extensión .CSS para indicarnos de que se trata, dentro de este archivo sólo se puede incluir sintaxis CSS, no se puede escribir nada de código HTML, por ejemplo:

```
P{
  Font-size:18pt;
  Font-family:arial;
  Font-weight:normal;
  Color:#ff0000
}

H1{
  Font-size:50pt;
  Font-Family:Arial;
  Text-decoration:Underline;
  Text-align:center;
  Background-color:Teal;
}
```

CLASES.

Las clases permiten crear definiciones de estilos que se pueden utilizar muchas veces, aplicándolos a diversas etiquetas.

Se pueden considerar tres tipos de clases que poseen una herencia de valores de los atributos y una jerarquía.

a) Clases generales: se pueden aplicar a cualquier etiqueta y se pueden definir bien en un archivo independiente de estilos o bien en la cabecera de una página entre `<Style>` y `</Style>`, la sintaxis es la siguiente:

```
Nombreclase{Text-decoration:underline;text-align:center;}
```

Esta clase se puede aplicar a cualquier etiqueta. Definida una clase general puede utilizarse con cualquier etiqueta HTML dentro del cuerpo utilizando el atributo CLASS, por ejemplo: <P Class="clase1">.

Nota: para evitar que un navegador que no reconozca la etiqueta Style presente el texto de la definición de clases, la definición de clases la haremos siempre dentro de un comentario.

b) Clases para un cierto tipo de etiqueta: dentro de una etiqueta (por ejemplo Hx) se pueden definir clases, cada una de estas clases puede tener un formato diferente. La definición de estas clases se puede hacer en el archivo independiente o en la cabecera entre <Style> y </Style>, por ejemplo:

```
H1.clase1 {Text-decoration:underline; text-align:right;}
H1.claseN1 {Text-decoration:underline; text-align:right}
```

Si el nombre de una clase de etiqueta coincide con el nombre de una clase general se heredan los atributos de la clase general para los cuales su valor no se especifica en la clase de etiqueta, sin embargo si dos atributos tienen distinto valor en la clase general y en la clase de etiqueta predomina el valor de la clase de etiqueta.

c) Estilo para una sola etiqueta en concreto; esto puede hacerse si al definir la etiqueta en el body especificamos el atributo ID.

Si tenemos una etiqueta que tenga un ID, podemos definir una clase para esa etiqueta de la siguiente manera: #ID {border-width:1px;border:solid;}. Este ID se puede establecer en la cabecera del documento entre <Style> y </Style> o en un archivo de estilos.

Nota: Las clases definidas en un archivo de estilos independiente predominan sobre las definidas en la cabecera de la propia página (Fallo de Internet Explorer 6.0).

Las etiquetas <DIV> y junto con las hojas de estilo y los atributos class e ID permiten extender la jerarquía y la herencia de la definición de clases y estilos a trozos del texto o a una sección de una página.

ESTILO DE LOS ENLACES.

Un enlace puede tener cuatro estados, no visitado, visitado, activo y con el ratón encima.

Se define el estilo de un enlace asignando los atributos a cada uno de sus estados, para ello, en la cabecera dentro de <Style> y </Style> escribimos lo siguiente:

A:link {atributos} → Se refiere a los enlaces no visitados.

A:visited {atributos} → Se refiere a los enlaces visitados.

A:active {atributos} → Se refiere a los enlaces activos.

A:Hover {Atributos} → Se define para el estilo de los enlaces cuando el ratón está encima.

Para que el enlace no esté subrayado escribiremos Text-Decoration:none.

ESTILO DE UN ENLACE CONCRETO.

Se establece dentro de la etiqueta A con el atributo Style:
...... En este caso todos los estados tendrán el mismo estilo.

ESTILO DE UN ENLACE MEDIANTE CLASES.

Podemos definir clases para los enlaces:

A.claseE:Link {Atributos}

A.claseE:Visited {Atributos}

A.claseE:Active {Atributos}

A.claseE:Hover {Atributos}

Para que se hereden esos caracteres utilizaremos en la etiqueta del enlace el atributo class:TEXTO.....

DEFINICIÓN FORMAL DE LAS HOJAS DE ESTILO CSS.

Una hoja de estilo css está formada por un conjunto de reglas css. Una regla css consta de dos partes:

1.- Un selector: establece la unión de un documento HTML y la hoja de estilos. Todas las etiquetas HTML son posibles selectores.

2.- Declaración: Tiene dos partes:

a) Una propiedad, determina la presentación (establece el formato) de un elemento (etiqueta) HTML, hay más de 50.

b) Un valor para la propiedad, la sintaxis es:
selector {propiedad1:valor;propiedad2:valor;}

Cada navegador tendrá una hoja de estilo predeterminado que presentará los documentos de una manera razonable.

ASIGNACIÓN DE VALORES A LAS PROPIEDADES.

Los navegadores para obtener el valor final de una propiedad realizan tres pasos:

1.- Asignar un valor especificado, para ello utilizará las siguientes reglas:

a) Si la cascada de estilos da como resultado un valor se utilizará ese valor.

b) Si no, si la propiedad es verdadera se utiliza el valor del elemento u etiqueta padre.

c) Si no, sí utiliza el valor inicial de la propiedad, el indicado en la definición de la propiedad. Los valores especificados pueden ser:

a) Absolutos: no se especifican con relación a otro valor.

b) Se especifican con relación a otro valor (%).

2.- Obtener el valor calculado o computado, los valores absolutos no necesitan ningún cálculo para obtener el valor calculado o computado, sin embargo los valores relativos hay que transformarlos para obtener el valor calculado o computado.

En la mayoría de los casos las etiquetas heredarán los valores computados, no obstante hay algunas propiedades o atributos de los que puede heredarse el valor especificado (por ejemplo Line-Head).

3.- Obtener el valor real, puede ocurrir que en ciertas circunstancias una aplicación de usuario no pueda utilizar el valor calculado, por lo que habrá que aproximarlos.

HERENCIA DE LAS ETIQUETAS ANIDADAS.

Si en una regla CSS se establece un valor para una propiedad o atributo de una etiqueta E_1 y en el código esa etiqueta anida a otra etiqueta E_2 esta hereda el valor del atributo establecido para E_1 si no existe una regla para E_2 que cambie el valor de dicho atributo, ejemplo:

```
Div {color:yellow}
....
..
<Div>
  <H1> Ejemplo</H1>
</Div>
```

VALOR POR DEFECTO DE UN ATRIBUTO O PROPIEDAD.

Para establecer una propiedad o atributo a un valor por defecto en un documento basta con darle un valor a dicho atributo o propiedad en un selector o etiqueta que anide a todas las demás etiquetas en HTML la etiqueta body sirve para esto.

PROPIEDADES NO HEREDADAS.

Algunas propiedades de estilos o atributos no se heredan del elemento padre al elemento hijo, por ejemplo la propiedad background no se hereda, no obstante el fondo del elemento padre se verá por defecto a través del elemento hijo.

VALOR INHERIT.

A cada propiedad o atributo puede asignársele el valor INHERIT, esto significa que para la etiqueta correspondiente la propiedad toma el mismo valor calculado o computado que el valor que tiene la propiedad en la etiqueta padre:

```
Body {Color:green}
```

```
H1 {Color:Inherit}
```

Esto es útil para aquellos atributos que no se heredan en otras etiquetas.

HOJAS DE ESTILO IMPORTADAS.

Se pueden importar hojas de estilo que existan en una determinada dirección URL; esto puede hacerse o bien en un archivo de hojas de estilo y entonces será la primera regla que se ponga o bien en la cabecera de una página Web dentro de <Style> y </Style>. La sintaxis es: @IMPORT URL(dirección url);.

SELECTORES CONTEXTUALES.

La herencia permite escribir menos al crear una hoja de estilos en lugar de establecer todas las propiedades de estilo se puede establecer primero los valores por defecto y luego las excepciones, ejemplo:

```
H2{Color:blue}
Em{Color:Fucsia}
```

↓

```
H2 Em{Color:Fucsia}
```

Los selectores contextuales están formadas por varios selectores simples que pueden ser etiquetas o claves separadas por espacios.

```
SN.....S(N-1).....S1 {atributo1:valor1;atributo2:valor2;}
```

Los valores de los atributos o propiedades que se establezcan sólo se refieren al último selector simple que se escriba.

Los selectores simples anteriores forman el llamado patrón de búsqueda.

Los valores establecidos sólo son efectivos si al escribir el código HTML la relación de descendencia (padre→hijo) con respecto al anidamiento entre los selectores implicados coinciden con el patrón de búsqueda, es decir, S1 está anidado en S2, S2 anidado en S3 y así hasta llegar hasta Sn. Ejemplo:

```
<Sn.....>
  <Sn-1....>
    .....
    ....
    <S1>
    </S1>
    ....
    .....
  </Sn-1>
</Sn>
```

Nota 1: Esto sólo funciona con las etiquetas que obligatoriamente tengan cierre

*Nota 2: El carácter * es el selector universal y respeta a todas las etiquetas.*

PSEUDO ELEMENTOS.

Se utilizan para hacer referencia a subpartes de elementos /etiquetas).

Pueden utilizarse como selectores en las hojas de estilo pero no existen en el código HTML si no que son insertados por el navegador bajo ciertas condiciones.

Para definir su comportamiento se puede utilizar una secuencia ficticia de etiquetas.

a) Pseudo elemento FIRST-LINE, por ejemplo: H5: FIRST-LINE{atributo:valor}.

Nota: a este pseudo elemento no se le pueden aplicar todos los atributos. Los atributos que se pueden aplicar son: font, color, background, word-spacing, letter-spacing, text-decoration, vertical-align, text-transform, line-height,clear.

b) Pseudo elemento FIRST-LETTER, se utiliza para las letras capitales, se escribe dentro de las reglas de estilo: *Etiqueta/identificador:First-letter{atributo:valor}*

```
#pc : first-letter {font-size:300%, text-transform:uppercase;}
#pc2 : first-letter {font-size:300%; float; text-transform:uppercase;}
```

Nota: a todo elemento que se le aplique el atributo float de valor left o right, lo convierte en flotante.

PSEUDO ELEMENTOS EN SELECTORES.

En un selector contextual los pseudo elementos sólo están permitidos al final del selector, los pseudo elementos pueden combinarse con clases y sólo puede especificarse un pseudo elemento en cada selector.

```
Body p: first-letter {color:red}
P.C1: first-letter {color:red}
```

También pueden combinarse varios pseudo elementos:

```
P{color:red; font-size:14pt}
P:first-letter{color:green; font-size:300%}
P: first-line{color:fuchsia}
```

Las propiedades establecidas en first-line serán heredadas por first-letter pero serán anuladas si las mismas propiedades se establecen en first-letter.

CONFLICTOS ENTRE REGLAS DE ESTILO.

Puede ocurrir que más de una hoja de estilo influya simultáneamente en la presentación de un elemento.

A veces surgirán conflictos. La resolución de conflictos se basa en que cada regla tiene un peso.

Tanto los autores como los lectores pueden influir sobre la presentación mediante hojas de estilo. Por defecto las reglas del lector tienen un peso menor que las reglas del autor y tanto las reglas del autor como las del lector prevalecen sobre los valores por defecto del navegador.

Cualquier regla especificada en un ahoja de estilo prevalece sobre las reglas especificadas en hojas de estilo importadas (@import).

Las reglas de una hoja de estilo importada prevalecen sobre las reglas de una hoja de estilo importada anteriormente en el código.

Declaración !Important: los autores de hojas de estilo pueden aumentar el peso de una declaración añadiendo !Important al valor de una propiedad o atributo por ejemplo:

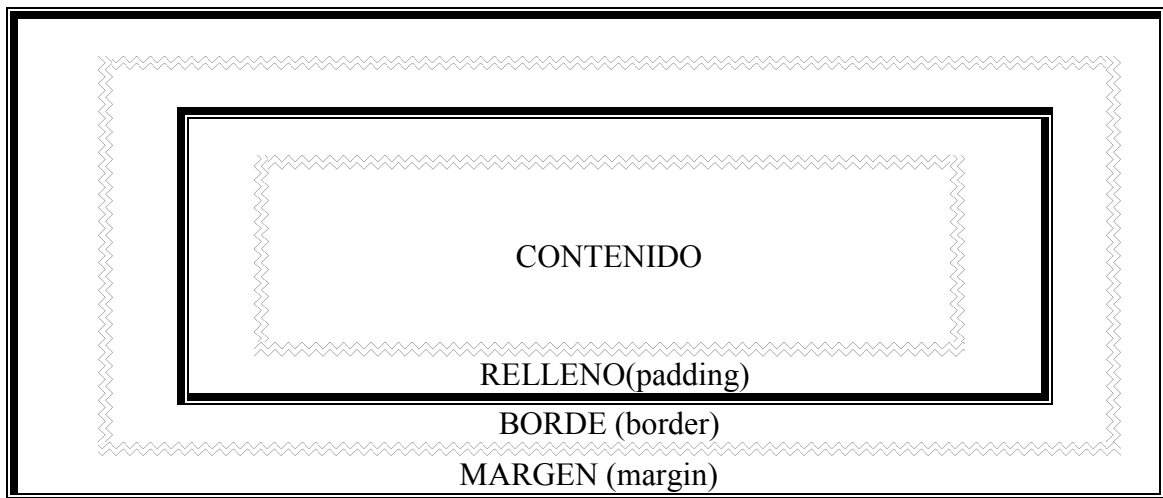
```
H1 {color:red !Important; font-size:12pt !Important; text-align:center;}
```

Una regla de lector con una declaración !Important prevalece sobre una regla de autor con una declaración normal.

Una regla de autor con una declaración !Important prevalece sobre una regla de lector con una declaración !Important.

MODELO DE CAJA.

Cada elemento de un documento genera una caja rectangular, cada caja posee un área de contenido en la cual puede haber texto, imágenes, etc.. y opcionalmente un padding o relleno, un área de borde y un área de margen.



- El background del contenido cubre el contenido y el padding.
- El border por defecto es transparente, para que se vea el border-style tiene que ser diferente a none que es en valor por defecto.
- Los cuatro bordes pueden tomar colores (color) estilos(style) y en ancho (width) diferentes entre si.

```
Border-top-style:solid  
Border-bottom-color:red  
Border-left-width:12px
```

CAPAS.

Las capas son zonas rectangulares que contienen código HTML el cual se muestra por encima de la página en ejecución. Para crear una capa se pueden utilizar las etiquetas Div o Span.

Una capa es una parte de una página Web que tiene un comportamiento independiente dentro de la ventana del navegador pues podemos colocarla en cualquier parte de la misma y moverla por ella independientemente además de poder darle un formato independiente del resto de la página.

Nota: en Netscape se construyen páginas con las etiquetas layer o ilayer pero no son reconocidas por Internet Explorer.

DEFINICIÓN DE UNA CAPA.

Se puede hacer de dos maneras:

1ª.- de una manera análoga a la declaración de una clase para una etiqueta con un ID.

```
#Cp1 {position:absolute;left:100px;top:50px;background:#777777;margin: 100px 30px 20px 10px; text-align:center}
```

```
.....  
.....  
<body>  
.....  
.....  
<div di="cp1">.....</div>
```

2ª .- utilizando el atributo style dentro de una etiqueta div:

```
<Div Style="position:absolute ;left:150px; top:360px; width:127px; height:41px; visibility:visible">
```

Atributos fundamentales para definir una capa:

- Position: establece como interpretará el navegador las coordenadas de posición y de tamaño de la capa, puede tener dos valores:

-Absolute, en este caso la capa se sitúa en las coordenadas que se indiquen independientemente del contenido del resto de la página, la posición se establece con respecto al vértice superior izquierdo de la página.

-Relative, en este caso la posición de la capa se verá afectada por el contenido de la página. La posición se establece con respecto al punto de la página en que se estaba escribiendo al comenzar la capa.

En general se usará el valor absolute para poder manejar la capa desde un lenguaje de script.

-Visibility: define si la capa es visible o no, puede tener los siguientes valores:

-Visible, la capa es visible.

-Hidden, la capa no es visible.

-Inherit, que es el valor por defecto, la capa hereda la visibilidad de la capa padre.

Si la capa no está anidada dentro de otra capa se supone que está en la capa documento, la cual siempre es visible.

-Left, establece la distancia al borde izquierdo absoluto o relativo (al borde izquierdo de la capa padre).

-Top, establece la distancia al borde superior de la capa padre.

-Width, establece la anchura de la capa.

-Height, establece la altura de la capa.

TEXTO SOBRE UNA IMAGEN.

Puede hacerse de dos maneras:

1.- colocando el texto en una capa transparente sobre la imagen, en este caso, las coordenadas de la capa han de ser las adecuadas para que el texto quede sobre la imagen.

```
<Div style="position:absolute; left:150px; top:360px; width:127px; height:41px; visibility:visible">
```

```
<H1><font face="arial, helvetica"> TEXTO </font></div>
```

```

```

- 2.- Otra forma es utilizar la propiedad background-image:url(imagen).
- 3.- En una capa primaria se establece una imagen insertada, y en una segunda capa estableceremos el texto que se superpondrá a la imagen.

Capa 1 → Imagen.

Capa 2 → Texto.

Nota: cuando se coloca una imagen de fondo es una capa si la imagen es mayor que la capa sólo veremos parte de la imagen y si la imagen es mas pequeña que la capa se repetirá en mosaico hacia la derecha y hacia abajo para cubrir toda la capa.

PROPIEDAD OVERFLOW O DESVORDAMIENTO.

Esta propiedad determina el comportamiento del navegador cuando el contenido de la capa tiene un tamaño mayor que la capa, por defecto el navegador agranda la capa para que quepa todo el contenido, si se especifica la propiedad overflow pueden dársele los siguientes valores:

-Visible: el efecto es el mismo que no especificar la propiedad, la capa se agranda para dar cabida a todo el contenido.

-Hidden: la capa mantiene su tamaño por lo que el contenido que no quepa será invisible.

-Scroll: la capa mostrará unas barras de desplazamiento para poder visualizar todo el contenido pero no modifica el tamaño de la capa.

Las barras de desplazamiento vertical y horizontal aparecen aunque el contenido quepa en las dimensiones de la capa, aunque en este caso aparecerán inhabilitadas.

-Auto: en este caso las barras de desplazamiento sólo aparecen si el contenido es mayor que la capa, sino, no aparecen.

PROPIEDAD Z-INDEX O PROFUNDIDAD DE LAS CAPAS.

Suele ocurrir que varias capas se solapen parcial o totalmente. Para establecer que capa queda por encima y cual queda tapada se utiliza la propiedad z-index, la cual recibe un valor numérico.

Por defecto las capas se apilan de tal manera que queda encima la ultima que se define en el código.

Si se establece la propiedad z-index la capa cuyo valor de z-index sea mayor estará por encima de las capas con un z-index de menor valor. Z-Index:10.

ALINEACIÓN DEL TEXTO DE UNA CAPA.

Por defecto el texto que se escriba dentro de una capa se alinea a la izquierda, la propiedad Text-align permite alinear el texto de la capa, puede tomar los valores left, right, center y justify. Además se pueden utilizar el resto de las propiedades como font-family; font-face; text-decoration; etc.

RECORTES (clipping).

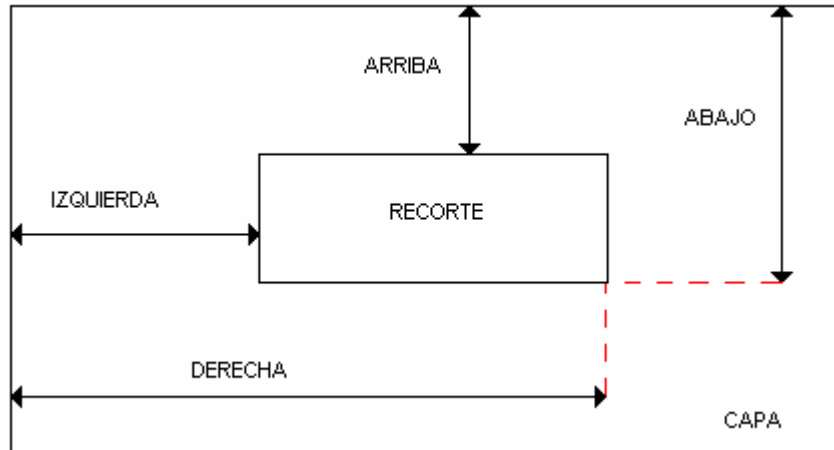
A veces interesa ver sólo una parte de una capa, esto se consigue mediante los recortes, para ello se utiliza la propiedad CLIP que permite establecer zonas rectangulares, la sintaxis es: *CLIP:rect (nºarriba,nºderecha,nºabajo,nºizquierda).*

-El número de arriba es la distancia entre el borde derecho del recorte y el borde izquierdo de la capa.

-El número de la derecha, establece la distancia entre el borde derecho del recorte y el borde izquierdo de la capa.

-Abajo es la distancia entre el borde inferior del recorte y el borde superior de la capa.

-Izquierdo es la distancia entre el borde izquierdo de la capa y el borde izquierdo del recorte.



Sólo el rectángulo correspondiente al recorte es visible, el resto es invisible.

FILTROS EN LAS CAPAS.

A las capas se les puede añadir efectos especiales llamados filtros, esto se realiza con la propiedad FILTER a la que se le asignan diferentes valores según el efecto que queremos conseguir.

a) Reflexión sobre el eje horizontal. Se hace con la propiedad filter asignándole el valor FLIPV(). FILTER:FLIPV().

b) Reflexión sobre el eje vertical. Se hace asignándole a FILTER el valor FLIPH(). Por ejemplo: FILTER:FLIPH().

c) Combinación de los dos anteriores. Simetría con respecto al origen de coordenadas. FILTER: FLIPH() FLIPV()

d) Transparencias. Se puede hacer que una capa sea más o menos transparente de forma que se pueda ver lo que hay debajo de ellas, se hace de la siguiente manera: FILTER: alpha (apacity=n°).

El valor del opacity puede variar desde 0 (totalmente transparente) hasta 100 (capa totalmente opaca).

e) Transparencias de un color. Se puede hacer que un color de la capa sea totalmente transparente, de forma que se vea lo que hay debajo de ese color, esto se hace : FILTER: chroma(color=color a transparentar).

f) Ondas. Se puede deformar las capas en ondas, para ello se utilizará: FILTER: WAVE (parámetros).

- f.1) Cortinillas: FILTER:WAVE (FREC=nºcortinillas).
- f.2) Sombreado a bandas: FILTER:WAVE (STRENGTH=nº). Cuanto mayor es el valor de strength más deformado aparecerá el contenido de la capa pero o se puede controlar el número de cortinillas que aparecen.
- f.3) Bandas desfasadas: FILTER:WAVE (PHASE=nº). El número puede ser entre 0 y 100, no se puede controlar el número de cortinillas.
- f.4) Claridad de las cortinillas: FILTER:WAVE (LIGHTSTRENGTH=nº). El número puede variar entre cero (no se ven las cortinillas) y cien (las cortinillas son negras).
- f.5) combinación de las anteriores: FILTER:WAVE(FREC=Nº, PHASE=Nº,..)
- g) Negativo de una capa. Permite invertir todos los colores de la capa, se hace con FILTER:INVERT ().
- Combinación de filtros. A una capa se le pueden aplicar varios filtros simultáneamente, se escriben separadas por espacios, por ejemplo: FILTER: INVERT() WAVE (STRENGTH=30, LIGHTSTRENGTH=80).

INSERCIÓN.

Son archivos externos al documento HTML creados con aplicaciones específicas que pueden transmitirse por la red y ejecutarse en los navegadores como una parte de la página Web, entre ellas se encuentran imágenes, sonidos, applets de java, videos, animaciones de Flash, marcos en línea...

Para que las inserciones se ejecuten correctamente es necesario que el sistema operativo tenga instalados unos plug-ins, los navegadores actuales incorporan la mayoría de los plugo-ins necesarios.

APPLETS DE JAVA.

Los applets de java son pequeños programas ejecutables escritos en Java que se pueden ejecutar desde e insertar dentro de una página Web, cuando el usuario cargue una página, el código Java será ejecutado por su navegador (si su navegador soporta Java, si no lo ignora).

El programa Java es un archivo de extensión .class que se guarda con los demás programas que forman el sitio Web.

Este es un archivo compilado, no es un archivo de texto. No obstante estos archivos no pueden ejecutarse directamente, están compilados para ejecutarse formando parte de un documento HTML.

Si bien el applet de java puede ser cualquier programa en Java no debe ser demasiado grande para evitar que tarde demasiado tiempo en cargarse.

Por otro lado Java distingue entre mayúsculas y minúsculas por lo que , al referirse al programa Java desde HTML, se escribirá el nombre del programa Java respetando las mayúsculas y minúsculas del nombre con el que haya sido guardado.

Una manera de insertar los applets de Java es mediante la etiqueta <applet>. El W3C ha desaprobado la etiqueta <applet>.

Inserción de un applet de Java. Lo primero es tener el archivo compilado de extensión .class del programa en Java, dentro del cuerpo de la página se escribirá:

```

<applet codebase="trayectoria_archivo" code="nombre archivo.class"
width="ancho ventana en que se ejecuta el programa Java" height="altura de la
ventana">
    <param name="nombre parámetro" value="valor del parámetro">
    <param name="speed" value="7">
    Contenido para navegadores que no soportan Java
</applet>

```

Nota: los applets se diseñan para ejecutarse en un área de un cierto tamaño, si el área reservada es mayor no habrá problemas, pero si el menor sólo se ve la parte del applet, por este motivo se utilizarán medidas absolutas (píxeles) y no relativas (%).

Otros parámetros que se pueden utilizar en la etiqueta applet pueden ser:

- Id o Name
- Alt, para que salga un texto alternativo.
- Hspace y Vspace, para establecer márgenes.
- Align, para alinear la ventana horizontalmente.

CONCEPTO DE SERVLETS.

Son módulos en java que se ejecutan en el servidor mostrando los resultados de la ejecución de los módulos en los navegadores de los clientes.

INSERCIÓN DE VIDEOS.

Se pueden insertar videos en diversos formatos por ejemplo .asx .asf .mpeg .avi, para utilizar videos en una página Web a través de Internet hay que tener una conexión con un ancho de banda elevado(al menor RDSI). Para crear un video se necesitan programas especiales.

Hay dos procedimientos para insertar videos:

1.- , en este caso al abrir la página Web se ejecuta automáticamente una sola vez del video. Al insertar el video la imagen se ejecuta al tamaño de la ventana por lo que si el tamaño no es proporcional al tamaño real de la imagen se deforman las imágenes.

2. Mediante la etiqueta <embed>; <embed src="video">. A esta etiqueta le podemos asignar los siguientes parámetros:

- loop, puede tomar dos valores, TRUE, significa que el video se ejecuta ininterrumpidamente y si es FALSE el video sólo se ejecuta una vez, el valor por defecto es FALSE.

-Autostart, puede tener dos valores, TRUE, el video se reproduce automáticamente al cargar la página y FALSE, para reproducir el video el usuario tiene que pulsar un botón en una barra de control que aparece debajo de la imagen, el valor por defecto es TRUE.

-Width, ancho de la ventana en que se ve el video.

-Height, alto de la ventana en que se ve el video.

-Hidden, que puede tener dos valores, TRUE, el video es invisible y FALSE, el video puede verse.

-No admite Hspace y Vspace.

Embed abre una ventana en la que se reproduce el video y debajo aparece un panel de controles suministrado por el sistema operativo con los cuales se puede iniciar, pausar, parar el video o modificar el volumen.

-Sonidos con Embed; `<embed src="sonido.wav" autostart="false">`.

ANIMACIONES DE FLASH.

Flash es un programa para realizar animaciones con el cual se consiguen efectos espectaculares fácilmente, las animaciones de flash se guardan en archivos de extensión .swf, los archivos utilizan un buen compresor y son de tamaño reducido. Para insertar estos archivos:

`<embed src="trayectoria archivo.swf">`, en los navegadores antiguos: `<embed src="archivo.swf" type="application/x-sockware-flash">`. Si el usuario no tiene instalado el plug-in de flash podemos que lo descargue automáticamente de la siguiente manera: `<embed src="archivo.swf" pluginpag="http://www.macromedia.com/shockwave/download/index.cgi?p1_pro_ver_sion=shockwave.flash">`.

INSERCIÓN DE MARCOS EN LÍNEA.

Permite insertar un documento HTML dentro del texto de otro documento HTML, para ello se utiliza la etiqueta `<IFRAME>` → `<Iframe src="página.html"></Iframe>`, otros atributos que se pueden poner son:

-Weight y Height como dimensiones de la ventana.

-Scrolling, que puede tomar tres valores:

-YES, se ven barras de desplazamiento.

-No, no se ven barras de desplazamiento.

-AUTO, si la página no cabe en la ventana se ven las barras de desplazamiento, pero si cabe no se ven.

EL documento incluido sigue siendo independientemente del documento principal.

-Frameborder, puede valor 0 o 1, si vale 1 la ventana tiene un borde y si vale 0 no tiene borde.

INSERCIÓN DE OBJETOS CON LA ETIQUETA <OBJET>.

-Imágenes, la sintaxis sería: `<objet data="imagen.gif" type="image/extensiónarchivo"></objet>` (no funciona en Internet explorer).

-Marcos en línea, la sintaxis es: `<objet data="página.html"...parámetros...>`

-Applets de Java, la sintaxis sería igual que con la etiqueta applet pero cambiando esta por `<objet>` y `</objet>`.

FUENTES DESCARGABLES.

Son archivos que contienen tipos de letras y que pueden descargarse a la vez que la página Web.

Los archivos de tipos de letras utilizados por Netscape tienen la extensión .PFR y los utilizados por Internet Explorer tienen extensión .EOT.

Ambos formatos son incompatibles. Veremos el caso de Internet Explorer.

El formato EOT es una variación del truetype en la cual se comprimen los archivos de definición de tipos.

CREACIÓN DE ARCHIVOS EOT.

Microsoft proporciona gratuitamente una herramienta llamada WEFT (herramienta de fuente para introducir en la Web) con la que se pueden crear los archivos EOT a partir de las fuentes normales de Windows, no obstante algunos tipos de fuentes no pueden convertirse.

ENLACE Y USO.

Supongamos que tenemos ya obtenido un archivo EOT por ejemplo president.eot, para enlazar este tipo de letra a una página Web escribiremos en esta la siguiente regla de estilo.

```
@font-face {  
    font-family:president;  
    src:url(president.eot);}
```