

UNIDAD 7 - DHTML Y EFECTOS MULTIMEDIA

7.1 - INTRODUCCIÓN.

El HTML Dinámico (DHTML – Dynamic HTML) es una herramienta que nos va a permitir añadir interactividad a las páginas Web. Se puede utilizar para crear animaciones, juegos, aplicaciones, para introducir nuevas formas de navegar a través de los sitios Web, etc. DHTML ofrece la ventaja de que no requiere ningún tipo de plug-in (programas que amplían la posibilidad del navegador, generalmente para realizar una función específica) para poder utilizarlo.

DHTML es una característica de Netscape Communicator 4.0, y Microsoft Explorer 4.0 y posteriores versiones de ambos, además de Firefox, Google Chrome y otros navegadores. Es tarea del navegador mostrar y manipular las páginas Web. La forma en que Firefox y Microsoft implementan DHTML es diferente. Por ello usar esta tecnología puede convertirse en tarea compleja ya que hay que conseguir que las páginas se visualicen en los dos navegadores.

Podemos decir que DHTML es una combinación de HTML, Hojas de estilo en cascada (CSS, Cascading Style Sheets) y JavaScript.

Los **Cascading Style Sheets (CSS)** son la base del HTML Dinámico. Ofrecen una forma de crear un conjunto de estilos que definen cómo se mostrarán los elementos de una página.

Los **Cascading Style Sheets Positioning (CSSP)** son una extensión de los CSS, y permiten un control a nivel de pixel sobre la localización de cualquier elemento de la

página. Las propiedades de los CSSP se aplican, normalmente, a una etiqueta `<DIV>..</DIV>`.

La manera de utilizar HTML Dinámico es mediante el uso de las marcas **DIV**. Dentro de estas marcas se puede colocar código HTML, como por ejemplo imágenes, tablas o formularios. La característica principal de ambos objetos es que poseen un conjunto de atributos que pueden ser modificables. Esto permite crear una infinidad de efectos como animaciones, menús, etc.

`<DIV>...</DIV>` Este elemento indica un bloque de documento que se debería tratar como una unidad lógica. A este elemento se le suele llamar de varias formas: "bloque DIV", "elemento DIV", "capa-CSS" o, simplemente "capa". La sintaxis de una etiqueta DIV es similar a otras etiquetas HTML: `<DIV>Esto es una etiqueta </DIV>`

Al aplicar CSS a una etiqueta DIV, podemos indicar en qué lugar de la pantalla se mostrará el trozo HTML, con qué color, con qué tamaño, etc... Para esto, daremos a la etiqueta DIV un identificador (`ID="identificador"`) para poder referirnos a él:

```
<DIV ID="minombre">
Esto es una etiqueta DIV con identificador
</DIV>
```

7.2 - DEFINICIÓN DE ESTILOS.

El propósito de las hojas de estilo es crear una presentación para un elemento particular o para un grupo de elementos. Los estilos se pueden definir de dos formas:

A - Directamente en una etiqueta de la siguiente forma:

```
<etiqueta STYLE="propieda1:valor;
.....;
propiedadN:valor"> ...
</etiqueta>
```

etiqueta es la etiqueta de HTML a la que vamos a dar una apariencia concreta (`<P>`, ``, `<I>`, `<A>`...). *STYLE* es el parámetro que indica que vamos a aplicar el estilo definido a la etiqueta en la que se encuentra. *propiedad* será la

característica de la etiqueta que queremos modificar (color, tipo y tamaño de letra, etc). Valor es el valor que damos a la propiedad (color rojo, tamaño 10, etc.)

```
<DIV id="saludo" STYLE="position:absolute; top:10; left:50; width:100; background-color:#AADDFF">
Hola Mundo!
</DIV>
```

En este ejemplo, podemos ver un objeto DIV el cual esta colocado absolutamente (position:absolute) 50 pixels desde el margen izquierdo de la pantalla (left:50) y 10 pixels (top:10) desde la parte superior de la pantalla. El color de fondo es azul claro (background-color:#AADDFF), y el ancho 100 pixels (width:200).

```
< HTML>
<head>
<title> Estilos</title>
</head>
<BODY>
<DIV id="saludo" STYLE="position:absolute; top:10; left:50; width:100; background-
color:#AADDFF">
Hola Mundo!
</DIV>
</BODY>
</HTML>
```

Ejemplo 46 – Estilos (I)

B - O bien se suelen definir para que se apliquen a todas las etiquetas del documento que deseemos. Para ello usamos las etiquetas <STYLE>...</STYLE> que se han de definir en la cabecera del documento (entre <HEAD>...</HEAD>) para asegurarnos de que se aplicarán a todas las etiquetas. La sintaxis es:

```
<STYLE TYPE="text/css">
Etiq1,...,EtiqN : {propiedad1:valor;...;propiedadM:valor}
...
Etiqb,...,Etiqz : {propiedad1:valor;...;propiedadS:valor} Otros: {propiedad1:valor;...;propiedadT:valor}
</STYLE>
```

Etiq1, Etiq2, ...: Son las etiquetas HTML a las que se da una apariencia concreta. Podemos aplicar el mismo estilo a varias etiquetas escribiéndolas separadas por comas. Otros: Hace referencia a las llamadas clases (CLASS - precedidas de punto (.))

que nos permitirán definir un estilo para asignárselo a los elementos que queramos.

También nos permitirá crear reglas de estilo, precedidas de #.

```
<STYLE TYPE="text/css">
#ejemplo {aquí van los estilos (posición, color, tamaño..)}
</STYLE>
```

```
<DIV ID="ejemplo">
Esto es una etiqueta DIV con cierto estilo
</DIV>
```

Partimos del ejemplo anterior, definimos una regla de estilo llamada #saludo con las características de color y posición definidas en el ejemplo. Definimos un párrafo con un ID igual al de la regla de estilo (<P ID="saludo">). A dicho párrafo se le aplicará el estilo definido. El código es el siguiente:

```
<HTML>
<head>
<title>Estilos</title>
<STYLE TYPE="text/css">
#saludo {position:absolute; top:10; left:50; width:100;background-color:#AADDFF}
</STYLE>
</head>
<BODY>
<P>Primera linea</P>
<P ID="saludo">Hola Mundo!</P>
<P>Segunda linea</P>
</BODY>
</HTML>
```

Ejemplo 47 – Estilos (II)

En el ejemplo anterior se creó una regla de estilo para un elemento con un atributo **ID="saludo"**. Como el atributo ID debe ser único, para poder asignar a varios elementos un mismo estilo utilizamos el atributo **CLASS**. En este ejemplo, definimos una clase llamada *miclase* con las características de color de fondo amarillo y ancho de 100; y se la asignamos a varios elementos **<P>**. El código es el siguiente:

```

<HTML><head>
<title>Estilos</title>
<STYLE TYPE="text/css">
#saludo {position:absolute; top:10; left:50; width:100;
background-color:#AADDFF}
.miclase {background-color:yellow; width:100}
</STYLE>
</head><BODY>
<P CLASS="miclase">Primera linea</P>
<P ID="saludo">Hola Mundo!</P>
<P CLASS="miclase">Segunda linea</P>
</BODY>
</HTML>

```

Ejemplo 48 – Estilos (III)

La siguiente tabla muestra una serie de propiedades de los estilos que suelen funcionar en los navegadores.

Propiedad	Función
position	Define cómo se posicionará la etiqueta DIV: "relative" indica que las coordenadas están especificadas con respecto a otro elemento, mientras que "absolute" significa DIV se colocará en las coordenadas señaladas
left	Posición izquierda de la etiqueta DIV (número de pixels desde el extremo izquierdo del navegador).
top	Posición superior de la etiqueta DIV (número de pixels desde el extremo superior del navegador)
width	Ancho de la etiqueta DIV. Cualquier cosa que se coloque en el interior de la etiqueta, se ajustará según este valor.
height	Altura de la etiqueta DIV.
clip	Define el rectángulo de recorte de la capa. Los valores que se utilizan para definirlo son los de las cuatro esquinas: arriba, derecha, abajo, izquierda. clip:rect(superior,derecho,inferior,izquierdo)
visibility	Determina si la capa será "visible", "hidden" (oculta) o "inherit" (hereda visibilidad, opción por defecto).
z-index	La profundidad de la etiqueta DIV.
background-color	Color de fondo de la etiqueta DIV.
layer-background-color	Color de fondo de la etiqueta DIV. Etiqueta para el antiguo Netscape.
background-image	Imagen de fondo de la capa.
layer-background-image	Imagen de fondo de la capa. Etiqueta para el antiguo Netscape.

Resumiendo, podemos asignar propiedades a la etiqueta DIV de dos formas:

En la propia etiqueta:

```
<DIV ID="ejemplo" STYLE="position: absolute;  
left:100; top:100; width:300>  
Estilo en etiqueta DIV  
</DIV>
```

Fuera de la etiqueta:

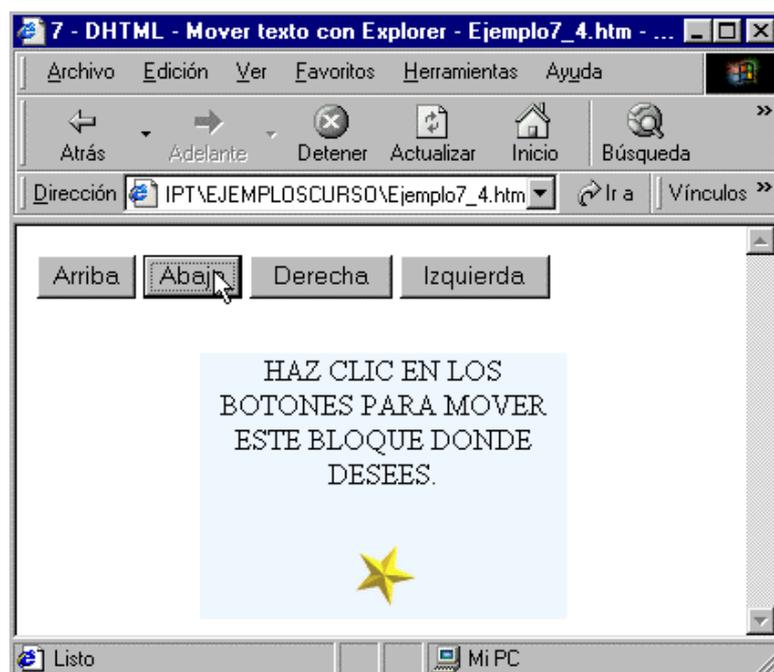
```
<STYLE TYPE="text/css">  
#ejemplo {position: absolute; left:100;  
top:100; width:300}  
</STYLE>
```

```
<DIV ID="ejemplo">  
Etiqueta DIV con estilo  
</DIV>
```

7.3 - TRABAJAR CON HTML.

7.3.1 - MOVER TEXTO E IMÁGENES.

A continuación se expone un ejemplo en el que 4 botones decidirán el movimiento de un texto y una imagen hacia arriba, abajo, a la derecha o a la izquierda. El ejemplo se muestra en la figura siguiente:



```

<HTML>
<HEAD>
<TITLE> Mover texto con Explorer </TITLE>
<STYLE TYPE="text/css">
#Parrafo {position:absolute; top:50; left:100; width:200;
height:100; background-color:aliceblue}
</STYLE>
<SCRIPT LANGUAGE="JavaScript">
function MoverTexto(d) {
if(document.all.Parrafo.style.pixelTop<=0)
document.all.Parrafo.style.pixelTop=50;
if(document.all.Parrafo.style.pixelLeft<=0)
document.all.Parrafo.style.pixelLeft=100;

switch(d) {
case "A": document.all.Parrafo.style.pixelTop -= 10; break;
case "B": document.all.Parrafo.style.pixelTop += 10; break;
case "D": document.all.Parrafo.style.pixelLeft += 10; break;
case "I": document.all.Parrafo.style.pixelLeft -= 10; break;
}
}
</SCRIPT>
</HEAD>
<BODY BGCOLOR=white>
<DIV ID="Parrafo" >
<P ALIGN="CENTER">HAZ CLIC EN LOS BOTONES PARA MOVER ESTE BLOQUE DONDE DESEES.</P>
<P ALIGN="CENTER"></P>
</DIV>
<TABLE BORDER=0>
<FORM NAME="FORMULARIO">
<INPUT TYPE=BUTTON VALUE="Arriba" onClick="MoverTexto('A');">
<INPUT TYPE=BUTTON VALUE="Abajo" onClick="MoverTexto('B');">
<INPUT TYPE=BUTTON VALUE="Derecha" onClick="MoverTexto('D');">
<INPUT TYPE=BUTTON VALUE="Izquierda" onClick="MoverTexto('I');">
</FORM>
</TABLE>
</BODY>
</HTML>

```

Ejemplo 49 – Mover texto

Las posiciones iniciales de la imagen y el texto (o del párrafo DIV) son: posición superior 50 y posición izquierda 100 (top:50; left:100). Al hacer clic en los botones se irá sumando o restando 10 a estas posiciones, según el botón pulsado.

En el ejemplo las posiciones superior (top) e izquierda (left) se inicializan cuando son menores o iguales a cero. No se controla la situación de que se superen la altura y el ancho del documento.

Se puede usar JavaScript para acceder y cambiar las propiedades de los elementos

CSS-P. La sintaxis difiere según se use Firefox o Explorer. En Firefox, la forma general para acceder a una propiedad CSS-P es: `document.NombreCapa.NombrePropiedad` o bien `document.layers["NombreCapa"].NombrePropiedad` . En Explorer:

`NombreCapa.style.NombrePropiedad` o bien:

`document.all["NombreCapa"].style.nombrePropiedad`

`document.all.NombreCapa.style.nombrePropiedad`

donde nombrePropiedad puede ser cualquier propiedad CSS-P, como `left`, `top`, `width`, `position`, `visibility`, etc...

Veamos el significado de estos objetos: `document.all.Parrafo.style.pixelTop` y `document.all.Parrafo.style.pixelLeft`

Objeto `style`: Representa las características actuales para todos los posibles estilos dados a un elemento. Algunas propiedades son:

Propiedades:	Función
<code>pixelHeight</code>	Contiene la altura del objeto en pixels.
<code>pixelWidth</code>	Contiene el ancho del objeto en pixels.
<code>pixelTop</code>	Contiene la posición superior del objeto en pixels.
<code>pixelLeft</code>	Contiene la posición izquierda del objeto en pixels.

Estas propiedades se aplicarán al estilo creado de nombre Parrafo.

El siguiente ejemplo usa el objeto `style` para que el tipo de fuente del documento sea "Verdana": `document.body.style.fontFamily = "Verdana"`

Objeto `all`: Devuelve una referencia a la colección de elementos contenidos por el objeto. La colección `all` incluye un elemento por cada etiqueta HTML válidas.

Para poder manipular la etiqueta **DIV** de nombre Parrafo utilizamos la siguiente expresión: `document.all.Parrafo`, al que después aplicamos las propiedades del objeto `style`. La altura de la ventana nos la da esta propiedad: `document.body.clientHeight` La anchura de la ventana: `document.body.clientWidth`

Para que funcione en Mozilla es necesario cambiar la función `MoverTexto()`:

```
function MoverTexto(d) {
if(document.Parrafo.top<=0)
document.Parrafo.top=50;
if(document.Parrafo.left<=0)
document.Parrafo.left=100;
switch(d) {
case "A": document.Parrafo.top -= 10; break;
case "B": document.Parrafo.top += 10; break;
case "D": document.Parrafo.left += 10; break;
case "I": document.Parrafo.left -= 10; break;
}
}
```

Los objetos `document.Parrafo.top` y `document.Parrafo.left` tienen la misma misión que `document.all.Parrafo.style.pixelTop` y `document.all.Parrafo.style.pixelLeft`. Para poder manipular la etiqueta **DIV** de nombre Parrafo utilizamos la siguiente expresión: `document.Parrafo.NombrePropiedad=valor` en la que `top` y `left` tienen la misma misión que `pixelTop` y `pixelLeft`. La altura de la ventana nos la da esta propiedad: `window.innerHeight` mientras que la anchura de la ventana `window.innerWidth`

En este ejemplo se moverá el texto sin necesidad de usar los botones. Inicialmente el texto está en las posiciones (5,5), si la posición Top supera la altura de la ventana se vuelve a inicializar a 5, si la posición Left supera la anchura también la inicializamos a 5. Para que empiece a moverse el texto nada más cargar el documento se incluye el evento `onLoad` asociado a la etiqueta `<BODY>` invocando a la función `MoverTexto()`. La función `SetTimeout`, espera 60 milisegundos para volver a llamar a la función para que el texto se mueva por la ventana.

```

<HTML>
<HEAD>
<TITLE> Mover texto sin botones </TITLE>
<SCRIPT LANGUAGE="JavaScript">
<!--
function MoverTexto() { document.all.Parrafo.style.pixelTop +=5;
document.all.Parrafo.style.pixelLeft +=5;

if(document.all.Parrafo.style.pixelTop > (document.body.clientHeight-50) )
document.all.Parrafo.style.pixelTop=5;
if(document.all.Parrafo.style.pixelLeft > (document.body.clientWidth-50))
document.all.Parrafo.style.pixelLeft=5;
setTimeout( "MoverTexto()",60);
}
//-->
</SCRIPT>

<STYLE TYPE="text/css">
#Parrafo {position:absolute; top:5; left:5; width:200; color:red }
</STYLE>
</HEAD>
<BODY BGCOLOR=#00FFFF onLoad="MoverTexto()">
<DIV ID="Parrafo" >
<P ALIGN="CENTER">ESTE TEXTO SE MUEVEEEEE!!!.</P>
<P ALIGN="CENTER"></P>
</DIV>
</BODY>
</HTML>

```

Ejemplo 50 – Mover texto sin botones

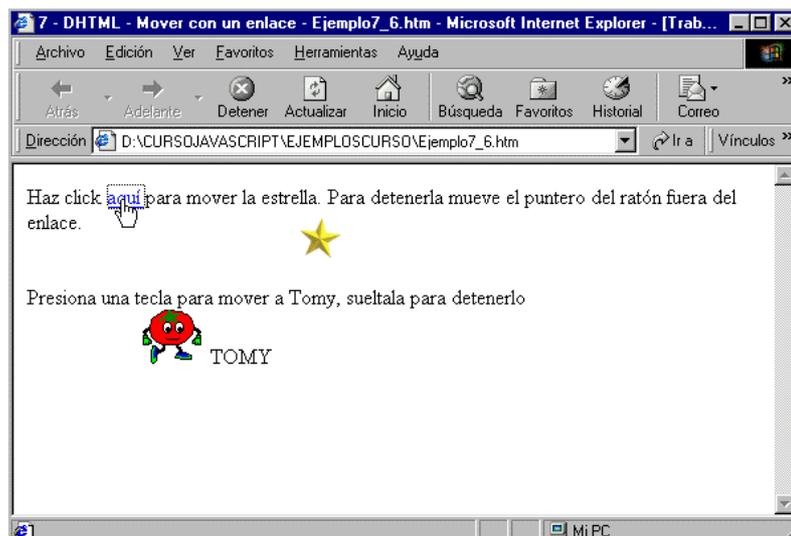


Para evitar los problemas derivados de las distintas formas de acceder a los elementos de una página Web desde Javascript, **se debe utilizar el método `document.getElementById`**. Este método permite obtener la referencia de un elemento a partir de su id y es compatible para todos los navegadores modernos.

```
<IMG src="foto.jpg" id="mifoto"/>  
<script type="text/javascript">  
foto=document.getElementById("mifoto");  
</script>
```

Una vez obtenida la referencia al elemento se pueden manipular y leer todas las propiedades del mismo. Es importante respetar el uso de mayúsculas y minúsculas a la hora de escribir correctamente el método `getElementById`.

En este ejemplo moveremos una imagen (la estrella) al hacer clic en un enlace (`onclick`); dejará de moverse cuando abandonemos el enlace con el puntero del ratón (`onmouseout`). Para mover otra imagen (tomy) pulsaremos una tecla (`onkeydown`), mientras la tecla está pulsada se moverá la imagen (sólo válido para Internet Explorer), al soltarla (`onkeyup`) se parará.



```

<HTML>
<HEAD>
<TITLE>7 - DHTML - Mover con un enlace - </TITLE>
<STYLE TYPE="text/css">
#Estrella {position:absolute; top:30; left:5}
#Tomy {position:absolute; top:110; left:5}
</STYLE>
<SCRIPT language=JavaScript>
corxEstrella=5;
corxTomy=5;
var clic = false; //variable que controla la pulsación del enlace
var mov_tomy=true; //variable que controla si se mueve o no a tomy
//cada vez que se pulse una tecla se ejecuta la función MoverTomy
document.onkeydown = MoverTomy;
//al soltar una tecla se ejecuta la función PararTomy
document.onkeyup = PararTomy;
function Mover(){
if (clic) { //se ha hecho clic en el enlace, se mueve la estrella
corxEstrella +=3
document.getElementById("Estrella").style.left=corxEstrella
setTimeout ("Mover()",30);
}
}
function MoverTomy(s){
if (mov_tomy) { //tecla pulsada, se mueve tomy
corxTomy +=3;
document.getElementById("Tomy").style.left=corxTomy;
}
mov_tomy=true;
}
function PararTomy(){
mov_tomy=false;
}
</SCRIPT>
</head>
<BODY > Haz click
<A href="javascript:void(null)"
onclick= "clic=true; Mover()"
onmouseout="clic=false"> aquí </A>para mover la estrella. Para detenerla mueve el puntero del
ratón fuera del enlace.
<DIV id="Estrella"><IMG src="ESTRELLA1.GIF" ></DIV>
<BR><BR><BR><P>Presiona una tecla para mover a Tomy, sueltala para detenerlo<P>
<DIV id="Tomy"><IMG src="tommy.gif" >TOMY</DIV>
</BODY>
</HTML>

```

Ejemplo 51 – Mover objetos con un enlace

7.3.2 -CAMBIAR EL COLOR DE TEXTO AL PASAR EL PUNTERO DEL RATÓN

Este ejemplo muestra cómo se puede cambiar el color de un texto al pasar el puntero del ratón por un link.



Este ejemplo sólo es válido para Internet Explorer. Se han definido tres capas (tres etiquetas `<div>` y `</div>`) de nombres: `enlace1`, `enlace2` y `enlace3`; contienen el texto que aparece bajo los enlaces. Se han definido dos funciones: `CambiarColor()` y `Restaurar()`, ambas reciben como parámetro el nombre de capa que deseamos cambiar el color. `CambiarColor(capa)` hace que el texto que hay en la capa (entre las etiquetas `<div>` y `</div>`) cambie al color rojo (red) y `Restaurar(capa)` hace que cambie al color inicial, es decir al negro (black). Para cambiar el color de la capa se utiliza la siguiente expresión: `document.all[capa].style.color = "red";`

Para invocar a las funciones se utilizan los eventos `onMouseOver` y `OnMouseOut`, el primero invoca a la función `CambiarColor()`, con el nombre de capa que cambiará de color; el segundo invoca a la función `Restaurar()`, con el nombre de capa que restaurará. Ejemplo:

```
onMouseOver="CambiarColor('enlace1')" onMouseOut="Restaurar('enlace1')">
```

La expresión: `var explorer = (document.all) ? true : false;` Comprueba el navegador, si es Internet Explorer, la variable `explorer` toma el valor `true`. A la etiqueta (`<A HREF>...`

`` se le ha aplicado el siguiente estilo: `A {TEXT-DECORATION: none}` que desactiva el subrayado de los enlaces. El código es el siguiente:

```

<html><head>
<title>Cambiar color de texto al pasar el ratón por un link</title>
<script language="javascript">
var explorer = (document.all) ? true : false; //comprobar navegador
function CambiarColor(capa){
if (explorer) document.all[capa].style.color = "red";
}
function Restaurar(capa){
if (explorer) document.all[capa].style.color = "black";
}
</script>
<STYLE type=TEXT/CSS>
A {TEXT-DECORATION: none}
</STYLE>
</head>
<body>
<center><table width="600" cellspacing="3" cellpadding="3">
<tr >
<td width="200">
<a href="unenlace.htm"
onMouseOver="CambiarColor('enlace1')" onMouseOut="Restaurar('enlace1')">
<b>OBJETOS</b></a>
</td>
<td width="200">
<a href="unenlace.htm"
onMouseOver="CambiarColor('enlace2')" onMouseOut="Restaurar('enlace2')">
<b>EVENTOS</b></a>
</td>
<td width="200"><a href="unenlace.htm" onMouseOver="CambiarColor('enlace3')"
onMouseOut="Restaurar('enlace3')">
<b>DHTML</b></a>
</td>
</tr>
<tr>
<td width="200">
<div id="enlace1" >Jerarquia de objetos, Propiedades y Métodos, Objetos predefinidos...</div>
</td>
<td width="200">
<div id="enlace2" >Lista de eventos más comunes: de ratón, de teclado, de formulario, etc..</div>
</td>
<td width="200">
<div id="enlace3" >DHTML, Mostrar, Ocultar y Mover capas, Menús deslizables, etc....</div>
</td>
</tr></table></center></form></body></html>

```

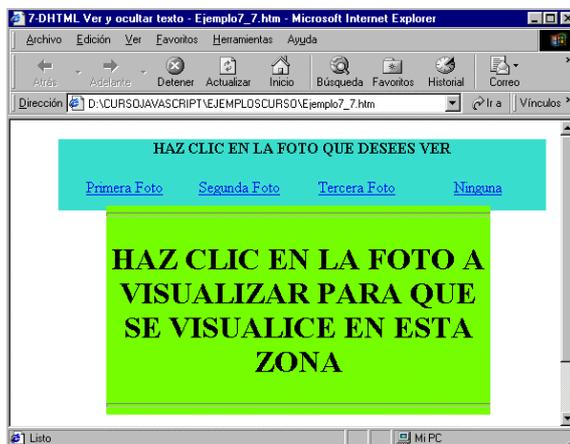
Ejemplo 52 – Cambiar color al texto

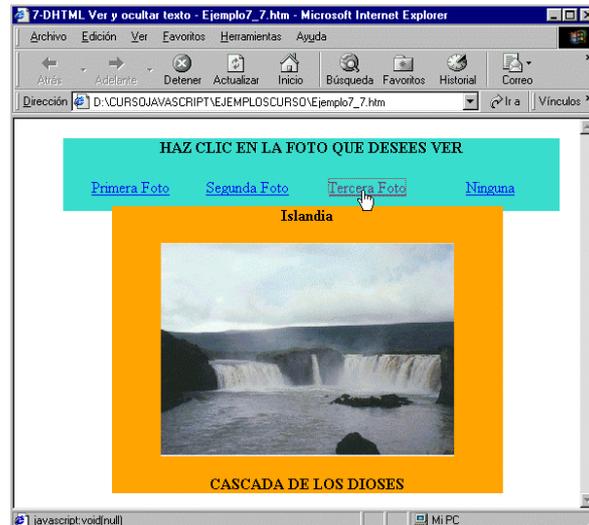
Si deseamos usar `getElementById` para que funcione en todos los navegadores, modificaríamos la función `CambiaColor()` del siguiente modo:

```
function CambiaColor(capa)
{
document.getElementById(capa).style.color="red";
}
function Restaurar(capa)
{
document.getElementById(capa).style.color="black";
}
```

7.3.3 - OCULTAR Y MOSTRAR TEXTO E IMÁGENES.

Este ejemplo muestra como se pueden visualizar y ocultar capas con texto o imágenes. Recordemos que una capa es lo que hay contenido entre las etiquetas <DIV> y </DIV>. Inicialmente aparece una pantalla con varios hipervínculos, y una capa (de nombre ninguna) indicando que hagamos clic en los mismos. Haciendo clic en cada hipervínculo se mostrará la capa correspondiente.





Veamos las capas que se han definido:

- capa0: capa que contiene los hipervínculos.
- capa1, capa2, capa3 y ninguna: son las capas que contendrán el texto e imágenes a mostrar. Tienen las mismas características de posición, ancho y alto, sólo varía el color de fondo. Inicialmente las capas 1, 2 y 3 están ocultas (`visibility:hidden`); la capa ninguna es la que aparecerá junto con la capa0.

Para desarrollar el ejercicio se han definido las siguientes funciones:

Mostrar(n): indica la capa a visualizar. Para visualizar una capa en `document.capa.visibility='show'`; ó `document.all.capa.style.visibility='visible'`;

Ocultar(): oculta las capas. Para ocultar una capa en `document.capa.visibility='hide'`; ó `document.all.capa.style.visibility='hidden'`;

```
<html>
<head>
<title>7-DHTML Ver y ocultar texto - Ejemplo7_7.htm</title>
<STYLE type=text/css>
#capa0 {
background-color: turquoise; layer-background-color: turquoise;
left: 50; position: absolute; width: 508; top:20; height: 75}
#capa1 {
background-color: lightblue; layer-background-color: lightblue;
visibility:hidden; left: 100; position: absolute; width: 400; top:90}
#capa2 {
background-color: olive; visibility:hidden;
layer-background-color: olive;
left: 100; position: absolute; width: 400; top:90
}
#capa3 {
background-color: orange ; visibility:hidden;
layer-background-color: orange ;
left: 100; position: absolute; width: 400; top:90
}
#ninguna {
background-color: lawngreen;layer-background-color: lawngreen;
left: 100; position: absolute; width: 400; top:90
}
</style>
<script language="JavaScript">
var Nav=false;
var IE=false;

if( navigator.appName == "Netscape" ) //detectar el navegador
Nav = true;
else IE = true;

function Mostrar(n){ //visualiza la capa seleccionada
switch (n) {
case 1:
if (Nav) document.capa1.visibility='show';
else document.all.capa1.style.visibility='visible';
break;
case 2:
if (Nav) document.capa2.visibility='show';
else document.all.capa2.style.visibility='visible';
break;
case 3:
if (Nav) document.capa3.visibility='show';
else document.all.capa3.style.visibility='visible';
break;
}
```

```

case 4:
if (Nav) document.ninguna.visibility='show';
else document.all.ninguna.style.visibility='visible';
break;
}
} function Ocultar() //oculta las capas
{
if (Nav) { document.capa1.visibility='hide'; document.capa2.visibility='hide';
document.capa3.visibility='hide'; document.ninguna.visibility='hide';
}
else {
document.all.capa1.style.visibility='hidden'; document.all.capa2.style.visibility='hidden';
document.all.capa3.style.visibility='hidden'; document.all.ninguna.style.visibility='hidden';
}
}
}
</script>
</head>
<body>
<div id="capa0" align="center" >
<P><B>HAZ CLIC EN LA FOTO QUE DESEES VER</P></B>
<table width="493">
<td width="147">
<p align="center">
<A href="javascript:void(null)" onclick="Ocultar();Mostrar(1)"> Primera Foto</p>
</a>
</td>
<td width="138">
<p align="center">
<A href="javascript:void(null)" onclick="Ocultar();Mostrar(2)">
Segunda Foto</p>
</a>
</td>

<td width="157">
<p align="center">
<A href="javascript:void(null)" onclick="Ocultar();Mostrar(3)">
Tercera Foto</p>
</a>
</td>
<td width="148">
<p align="center">
<A href="javascript:void(null)" onclick="Ocultar();Mostrar(4)">
Ninguna</p>
</a>
</td>
</table>
</div>

<div id="capa1" align="center" >
<b>Islandia</b>
<p></p>
<B>LAGUNA GLACIAR</B>
</div>

```

```
<div id="capa2" align="center" >
<b>Islandia</b>
<p></p>
<B>CAÑON DEL JOKULSA</B>
</div>

<div id="capa3" align="center" >
<b>Islandia</b>
<p></p>
<B>CASCADA DE LOS DIOSES</B>
</div>

<div id="ninguna" align="center" >
<HR SIZE=5>
<center><h1>HAZ CLIC EN LA FOTO A VISUALIZAR PARA QUE SE VISUALICE EN ESTA
ZONA</center></h1>
<HR SIZE=5>
</div>
</body>
</html>
```

Ejemplo 53 – Manipulación de capas