



MINISTERIO
DE EDUCACIÓN
Y CIENCIA

SECRETARÍA GENERAL
DE EDUCACIÓN
Y FORMACIÓN PROFESIONAL

DIRECCIÓN GENERAL
DE EDUCACIÓN,
FORMACIÓN PROFESIONAL
E INNOVACIÓN EDUCATIVA

CENTRO NACIONAL
DE INFORMACIÓN Y
COMUNICACIÓN EDUCATIVA

Edición HTML

Estilos CSS
Menús



SERVICIO DE
FORMACIÓN DEL
PROFESORADO

C/ TORRELAGUNA, 58
28027 - MADRID

Índice de contenido

Menú horizontal simple.....	3
Menú vertical simple.....	4
Menú horizontal desplegable.....	5
Menú horizontal desplegable en líneas.....	7
Menú vertical con cambio de alineación.....	7
Menú vertical desplegable.....	8
Menú vertical flotante.....	8
Menú vertical con flechas generadas.....	9
Mapa con imágenes de sustitución.....	9
Menú vertical con aviso de visitado.....	10

MENÚ HORIZONTAL SIMPLE

[Vínculo a sitio 1](#) [Vínculo a sitio 2](#) [Vínculo a sitio 3](#) [Vínculo a sitio 4](#)

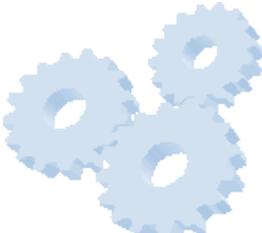
La idea base es eliminar las marcas de los elementos de lista poniendo el valor `list-style-type:none` y hacerlos flotar a la izquierda.

Para evitar que otros listados hereden los estilos de forma automática creamos una clase a la que llamamos `menu`. (Lo hacemos como clase en lugar de como indicador por si quisiéramos que en otro punto de la página apareciera otro listado con las mismas características).

Con este esquema base el resto de reglas se refiere a la apariencia gráfica de los elementos de la lista que sean enlaces: márgenes para separarlos, colores, alineación y bordes esencialmente.

Una vez definidas las características básicas comunes se especifican las particulares para las pseudoclasses `:visited` `:hover` y `:active` que provocarán los cambios de apariencia visual en función del estado del enlace.

```
<style type="text/css">
<!--
/* CSS base obtenido de los tutoriales de www.alsacreations.com/articles */
.menu {
height: 22px; /* necesario para que se produzca salto tras el menú */
}
.menu ul {
list-style-type: none;
width: 100%; /* ajuste para Opera */
}
.menu li {
float: left;
}
.menu a {
margin: 0 2px;
width: 120px;
height: 20px;
display: block;
text-align: center;
border: 1px solid gray;
text-decoration: none;
color: #000;
background-color: #fff;
}
.menu a:visited {
color: rgb(204, 102, 204);
background-color: rgb(221, 221, 221);
}
.menu a:hover {
background-color: rgb(204, 204, 204);
}
.menu a:active {
background-color: gray;
color: #fff;
}
-->
</style>
```



MENÚ VERTICAL SIMPLE

En este caso la situación es muy similar al menú horizontal simple. Las principales variaciones que encontramos son:

Vínculo a sitio 1

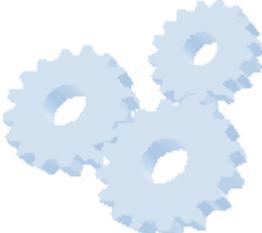
Vínculo a sitio 2

Vínculo a sitio 3

Vínculo a sitio 4

- Los elementos de la lista no flotan a la izquierda como hacían antes sino que siguen el flujo normal aunque se muestran como elementos en bloque para poder aplicarle los formatos gráficos.
- Debido al flujo normal de la lista no ha hecho falta establecer la altura del contenedor en que incluimos el menú porque se calcula correctamente como suma de los diferentes elementos
- En lugar de introducir un margen horizontal en los elementos establecemos un margen inferior para separar verticalmente.

```
<style type="text/css">
<!--
/* CSS base obtenido de los tutoriales de www.alsacreations.com/articles */
.menu {
    float:left;
    width: 180px;
    margin-right:10px;
}
.menu ul {
    list-style-type: none;
}
.menu li {
    margin-bottom: 5px;
}
.menu a {
    margin: 0 2px;
    width: 120px;
    height: 20px;
    display: block;
    text-align: center;
    border: 1px solid gray;
    text-decoration: none;
    color: #000;
    background-color: #fff;
}
.menu a:visited {
    color: rgb(204, 102, 204);
    background-color: rgb(221, 221, 221);
}
.menu a:hover {
    background-color: rgb(204, 204, 204);
}
.menu a:active {
    background-color: gray;
    color: #fff;
}
-->
</style>
```



MENÚ HORIZONTAL DESPLEGABLE

Menu 1	Menu 2	Menu 3	Menu 4
			Submenú 4.1 Submenú 4.2 Submenú 4.3

Este menú requiere que esté activado Javascript.

Los menús, debido a la utilización de un z-index:100 se superponen al contenido de la página al desplegarse.

Javascript

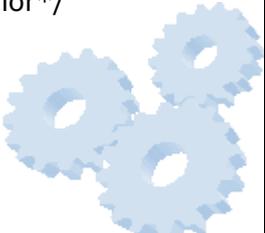
Lo situamos en la cabecera y se encarga de mostrar u ocultar los submenús:

```
<script type="text/javascript"><!--  
function mostrarsubmenu(id) {  
var d = document.getElementById(id);  
for (var i = 1; i<=10; i++) {  
    if (document.getElementById('smenu'+i))  
        {document.getElementById('smenu'+i).style.display='none';}  
    }  
    if (d) {d.style.display='block';}  
    }  
//-->  
</script>
```



Código CSS comentado

```
/*Eliminamos los identificadores de item y ajustamos margen y relleno*/  
dl, dt, dd, ul, li {  
margin: 0;  
padding: 0;  
list-style-type: none;  
}  
  
/*Creamos el identificador menu y lo situamos el menú en la zona superior*/  
#menu {  
position: absolute;  
top: 0;  
left: 10px;  
z-index:100;  
width: 100%; /* ajuste para Opera */  
}
```

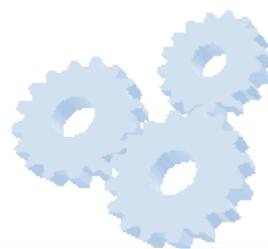


```
/*Aplicamos formato a los diversos componentes de las listas que se encuentren dentro
de #menu */
#menu dl {
float: left;
width: 12em;
}
#menu dt {
cursor: pointer;
text-align: center;
font-weight: bold;
background: #ccc;
border: 1px solid gray;
margin: 1px;
}
#menu dd {
display: none;
border: 1px solid gray;
}
#menu li {
text-align: center;
background: #fff;
}
/*Aplicamos formato a los enlaces que pertenezcan a elementos de listas que se
encuentren dentro de #menu */

#menu li a, #menu dt a {
color: #000;
text-decoration: none;
display: block;
height: 100%;
border: 0 none;
}
#menu dt a {
text-align:center;
}

/*Aplicamos pseudoclasas para formato para enlaces visitados y para el paso de ratón
sobre enlace dentro de #menu */
#menu li a:visited, #menu dt a:visited {
color: #eee;
}
#menu li a:hover, #menu dt a:hover {
background: #eee;
color: #000;
}

```

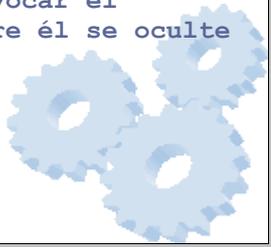


Observa que no tenemos una única lista de definición, sino que cada término de definición y sus datos constituyen una lista dl independiente. Haciéndolo así cada lista flota a la izquierda y permite que la siguiente se coloque a su derecha tal como especificamos al crear las reglas CSS.

```

<dl>
  <dt onmouseover="javascript:mostrarsubmenu('smenu1');">Menu 1</dt>
  <dd id="smenu1" onblur="javascript:mostrarsubmenu();">
    <ul>
      <li><a href="http://algun.sitio.edu">Submen&uacute; 1.1</a></li>
      <li><a href="http://algun.sitio.edu">Submen&uacute; 1.2</a></li>
      <li><a href="http://algun.sitio.edu">Submen&uacute; 1.3</a></li>
      <li><a href="http://algun.sitio.edu">Submen&uacute; 1.4</a></li>
      <li><a href="http://algun.sitio.edu">Submen&uacute; 1.5</a></li>
      <li><a href="http://algun.sitio.edu">Submen&uacute; 1.6</a></li>
    </ul>
  </dd>
</dl>
<dl>
  /* Aunque el siguiente menú no tiene submenús necesitamos invocar el
  javascript sin darle un valor para que al pasar el ratón sobre él se oculte
  el menú que estuviera abierto */
  <dt onmouseover="javascript:mostrarsubmenu();">
    <a href="http://algun.sitio.edu">Menu 2</a></dt>
</dl>
...

```



MENÚ HORIZONTAL DESPLEGABLE EN LÍNEAS



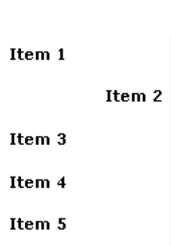
Este menú requiere que esté activado Javascript.

Alguna opción de menú podría no desplegar opciones funcionar como enlace directo a otra página. (Caso del Elemento menú 2 del ejemplo)

Se ha introducido la propiedad `display:table` para los selectores `dt` del menú para que se muestre correctamente en Opera.

También se ha añadido la pseudoclase `:visited` para introducir diferenciación de color en los enlaces visitados

MENÚ VERTICAL CON CAMBIO DE ALINEACIÓN



En este caso el efecto del menú es el cambio de alineación de los ítems y la modificación del color de la línea de delimitación inferior

MENÚ VERTICAL DESPLEGABLE

Menú 1
Menú 2
Submenú 2.1 Submenú 2.2 Submenú 2.3
Menú 3
Menú 4

En este caso nos encontramos con un menú en el que, al pulsar sobre cada una de las opciones nos llevará al destino especificado o bien abrirá un submenú. En ambos casos la pulsación sobre el menú principal cerrará el submenú que pudiera estar abierto.

En el ejemplo el primer elemento del menú sería un enlace directo mientras que los otros tres disponen de submenús desplegables.

Las acciones que se han descrito se apoyan en un `javascript` que se carga automáticamente con la página y se ocupa de realizar la tarea de apertura y cierre de los submenús.

Las reglas CSS se encargan de la apariencia de los diversos elementos y, para construirlos en el código HTML se ha utilizado una lista de definiciones: cada término de definición es un ítem del menú principal, mientras que el submenú son los datos de la definición que están constituidos por una lista que enumera cada uno de los submenús. Como ejemplo, además de consultar el código de la página, tienes a continuación el código correspondiente al cuarto ítem del menú principal:

```
<dt onclick="javascript:mostrarsubmenu('smenu4');">Menú 4</dt>
<dd id="smenu4">
  <ul>
    <li><a href="http://algun.sitio.edu">Submenú 4.1</a></li>
    <li><a href="http://algun.sitio.edu">Submenú 4.1</a></li>
  </ul>
</dd>
```

Como puedes ver, el término de definición contiene una llamada `javascript` que se activa al hacer clic que es la encargada de que se muestre el submenú.

Si reflexionamos sobre la estructura que se ha utilizado nos damos cuenta de que no es descabellada la utilización de una lista de definiciones, puesto que desde el punto de vista semántico sí que se podría interpretar un menú con submenús desplegables como tal.

Por la propia apariencia visual del menú su utilización más habitual será en páginas cuya estructura esté basada en marcos, bien sea en marcos puros o en marcos internos (`iframe`).

MENÚ VERTICAL FLOTANTE

Menu 1	
Menu 2	
Menu 3	Submenú 3.1
Menu 4	Submenú 3.1 Submenú 3.1 Submenú 3.1 Submenú 3.1 Submenú 3.1

Este menú requiere que esté activado Javascript.

Alguna opción de menú podría no desplegar opciones y funcionar como enlace directo a otra página. (Caso del Elemento menú 1 del ejemplo)

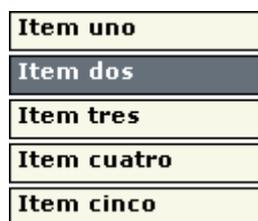
Los menús, debido a la utilización de un `z-index:100` se superponen al contenido de la página al desplegarse.

Por la propia estructura del menú su utilización más habitual será en páginas cuya estructura esté basada en marcos, bien sea en marcos puros o en marcos internos (`iframe`).

Al utilizar este menú en páginas con estructura real de marcos habría que tener en cuenta la anchura total del menú (elementos principales + elementos flotantes) para establecer el ancho del marco que lo contiene. Esto nos obligaría a revisar el código para establecer las

medidas del menú en píxeles.

MENÚ VERTICAL CON FLECHAS GENERADAS



La utilización de bordes gruesos en elementos de ancho y alto iguales a 0 provoca un curioso efecto que nos permite obtener zonas del color del borde con inclinaciones de 45 o 135° según el lado en que se produzcan al combinar los bordes izquierdo o derecho de un color con el superior y/o inferior de otro color. Gracias a esta combinación se ha obtenido la "flecha" roja que aparece al pasar el ratón sobre cada imagen.

Por si te parece mágico lo de las flechas observa atentamente el siguiente fragmento para ver de dónde salen esas inclinaciones de 45 o 135°.

```
<div style="border-style: solid; border-color:
rgb(255, 0, 0) rgb(255, 255, 0) rgb(0, 0, 255)
rgb(0, 255, 0); border-width: 8px;">
```



En el segundo caso los colores y el grosor del borde son iguales pero el ancho y el alto del elemento son 0

MAPA CON IMÁGENES DE SUSTITUCIÓN



Es sorprendente cómo se puede lograr un menú tan vistoso. La idea base es trabajar con listas mostradas en bloque. Se utilizan imágenes de sustitución: una imagen como fondo para los enlaces y otra diferente cuando el ratón pasa por encima. Consulta la documentación HTML para observar el efecto.

MENÚ VERTICAL CON AVISO DE VISITADO



En este caso el efecto que se persigue es la aparición de un rótulo que indica que ya se ha visitado el enlace.

Para comprobarlo puedes pulsar el primer enlace y luego la flecha de vuelta atrás de tu navegador.

Como estás viendo hemos dejado un importante margen en la zona derecha de los menús porque es necesario para dejar espacio a los rótulos que aparecerán. También habrá que tener en cuenta este detalle

cuando este tipo de menús se utilice en una estructura de marcos.

En esta ocasión no hemos creado una lista, sino una serie de contenedores encerrados dentro de otro contenedor genérico que tiene fondo gris. La idea en la que se basa el menú es que cada uno de los contenedores tiene el enlace y un fragmento en línea tal como puedes ver a continuación:

```
<div class="box">
<a href="http://www.cssplay.co.uk/menus/visitedmenu.html">Menú 1
<span>VISTO</span>
</a>
</div>
```

Para hacer que el fragmento en línea aprovechemos las características de herencia y utilizamos la propiedad CSS `display:none` para que cualquier elemento `span` que sea descendiente de un enlace que, a su vez, sea descendiente del contenedor que hemos nombrado con el identificador `menu`, no se muestre.

```
#menu a span { display:none; }
Bastante más simple de explicar con código CSS que con lenguaje humano ;-)
```

Lo único que nos queda es añadir la pseudo clase `:visited` a los enlaces para que los fragmentos en línea `span` que sean descendientes de un enlace visitado que, a su vez, sea descendiente del contenedor que hemos nombrado con el identificador `menu`, se muestren con la apariencia que indicamos con las reglas CSS. Además especificaremos un índice de profundidad alto para que se superponga a otros elementos que pudieran quedar por debajo.

```
#menu a:visited span {
display:block;
position:absolute;
top:0;
left:11em;
width:5em;
font-size:0.7em;
color:#c00;
border:1px solid #c00;
text-align:center;
z-index:50;
}
```

El resto de reglas CSS no merecen explicación especial ya que su función es la común de indicar el aspecto de los diversos elementos de la página.