

Realizzare un menu a tendina con DHTML

di Francesco Sblendorio

Chi naviga da tempo, avrà sicuramente notato una certa evoluzione sia nell'estetica che nelle funzionalità delle pagine Web presenti in rete. Questa evoluzione, resa possibile grazie all'uso intensivo di DHTML (ovvero HTML + CSS + Javascript), tende a rendere le pagine Web sempre più interattive, rendendole simili alle applicazioni native che vediamo girare sui nostri desktop. In questo articolo vedremo come sia possibile implementare la funzionalità dei menu a tendina all'interno dei nostri siti. In tal senso è da considerare una naturale "continuazione" dell'articolo [1] pubblicato su Login 31, che mostrava come implementare un'altra funzionalità, quella del drag'n'drop. Tutto il codice e gli esempi relativi a questo articolo sono reperibili presso l'indirizzo <ftp://ftp.infomedia.it/pub/Login/listati>

Gestione della compatibilità con i vari browser

Qui vale quanto detto in [1]: infatti utilizzeremo lo stesso file *crossbrowser.js*, che si occupa di riconoscere il browser sul quale si sta visualizzando la pagina. Come si sa, DHTML è una "babele" in quanto ogni browser lo interpreta a modo suo. Alla fine dell'esecuzione di questo piccolo script, avremo a disposizione tre variabili booleane (0/1) che ci indicheranno su quale browser stiamo lavorando:

- *isIE* indica che il browser utilizzato è Internet Explorer 4.x o superiore;
- *isN4* indica Netscape Navigator 4.x;
- *isN6* indica Netscape Navigator 6.x e più in generale tutti i derivati di Mozilla (vedi [3]).

Inoltre, questo script ci mette a disposizione varie funzioni; ne utilizzeremo tre:

- *obj('nomeLayer')*, restituisce un riferimento ad un layer (vedi [1]);
- *hideLayer(rifLayer)*, nasconde il layer identificato dal riferimento fornito;
- *showLayer(rifLayer)*, rende visibile il layer identificato dal riferimento fornito.

Ancora due note sulla compatibilità: la prima riguarda Netscape Navigator 4.x, all'interno del quale le voci del menu saranno visualizzate senza il bordo; questo è dovuto ad un problema di questo browser di cui ci occuperemo più avanti. L'altra nota riguarda Opera 5.x: i menu vengono visualizzati correttamente anche con questo browser, a patto di impostare l'opzione *File/Preferences/Connection/Browser-identification* al valore "Identify as Opera".

Il menu a tendina è un elemento onnipresente in ogni GUI che siamo abituati ad usare. Vediamo come implementare questa funzionalità all'interno di una pagina Web utilizzando DHTML

Come inserire il menu nelle pagine

Per inserire la funzionalità del menu a tendina in una pagina HTML basta inserire nella sezione `<HEAD>...</HEAD>` le seguenti righe:

```
<script type="text/javascript" src="
    crossbrowser.js"></script>
<script type="text/javascript" src="
    menu-settings.js"></script>
<script type="text/javascript" src="menu-text.js"></script>
<script type="text/javascript" src="menu-code.js"></script>
```

Il significato del primo file l'abbiamo spiegato nel paragrafo precedente. Occupiamoci ora degli altri tre file.

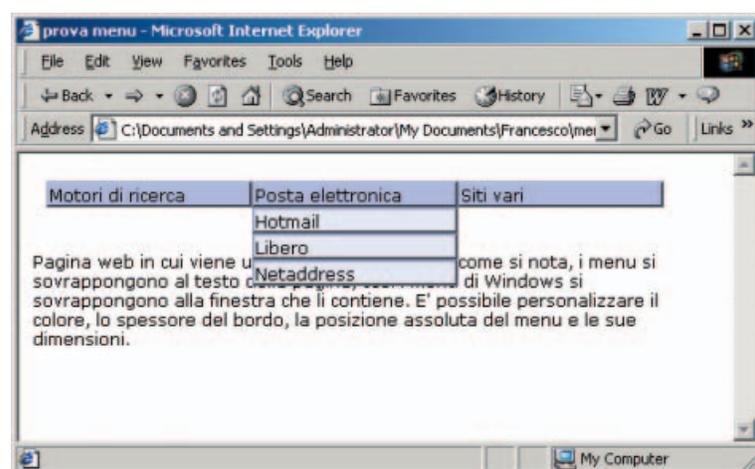
Il primo è *menu-settings.js*, che contiene impostazioni generali sull'aspetto dei menu (come il font utilizzato, le coordinate *x* ed *y*, ecc). Ne vediamo un esempio nel **Listato 1**, in cui ogni variabile impostata è adeguatamente commentata.

Il secondo è *menu-text.js*, che contiene il testo vero e proprio delle voci del menu a tendina. Un esempio è presente nel **Listato 2**: viene assegnato la variabile *menu*, un array che contiene stringhe ed array (ricordiamo infatti che in javascript un array può contenere elementi di tipo eterogeneo).

Definiamo questo array: gli elementi di indice 0, 2, 4, 6... contengono i nomi delle voci principali del menu, mentre gli elementi di indice 1, 3, 5, 7... contengono l'array del

FIGURA 1

Come appare il menu in Internet Explorer 5



LISTATO 1

Il file "menu-settings.js"

```
// File "menu-settings.js"
// Regola le proprietà di visualizzazione

// Coordinate iniziali del menu
startx=20;
starty=20;

rowLength=150; // Larghezza della voce del menu
rowHeight=15; // Altezza della voce del menu
borderWidth=2; // Spessore bordo

fontName="verdana,arial,Helvetica"; // Tipo carattere
fontSize="12"; // Dimensione carattere
nav4FontSize="2"; // Dimensione car. con Netscape 4

borderColor="#ccccee"; // Colore bordo
mainColor="#000000"; // Colore voci menu principale
itemColor="#000000"; // Colore voci sottomenu
mainBgColor="#c0c0ff"; // Colore sfondo menu principale
itemBgColor="#eeeeff"; // Colore sfondo sottomenu
```

sottomenu relativo; questi sotto-array hanno una struttura molto semplice: affiancano alla stringa da visualizzare nel menu, l'URL associato al clic.

Il terzo file javascript è *menu-code.js* (vedi Listato 3), che contiene il codice vero e proprio. Andiamo subito ad esaminarlo.

Le variabili globali utilizzate

Il file *menu_code.js* utilizza una serie di variabili globali, alcune utilizzate come flag di controllo, altre utilizzate per memorizzare risultati di calcoli intermedi. Esaminiamo le prime, che sono in pratica tre flag:

- *currentMenu* (intera): contiene l'indice (a partire da zero) del menu correntemente visualizzato; se non è visualizzato nessun menu, contiene -1;
- *dropdown* (booleana): contiene *true* se è stato attivato un menu, *false* altrimenti;
- *justDropped* (booleana): contiene *true* se è stato appena attivato un menu, *false* altrimenti; questo flag, apparentemente strano, serve per far sparire i sottomenu quando l'utente clicca al di fuori dell'area dei menu; vedremo meglio questo aspetto in seguito.

Altre variabili degne di nota sono:

- *cellWidth*, il cui valore è calcolato in base al browser utilizzato (uguale o diverso da Explorer);
- *borderWidth*, inizializzata nel file *menu-settings.js*, ma impostata a 0.5 nel caso di Netscape 4.x (infatti, in

questo caso il bordo sarà una sottile linea intorno alle voci del menu).

Generare dinamicamente elementi di una pagina: problematiche

Per visualizzare i menu dobbiamo leggere l'array "menu" e scrivere in qualche modo nella pagina. Per far ciò utilizzeremo il noto metodo *document.write()*, che però presenta qualche problema nel caso di Netscape 4.x.

Bisogna scrivere sulla pagina una serie di elementi <DIV>: ognuno di essi rappresenta un *layer* da posizionare (tramite l'attributo *style="..."*) e sul quale scrivere una singola voce di menu.

Quando questo tag è inserito direttamente in un documento HTML, viene visualizzato correttamente da Netscape 4.x (anche se con qualche limitazione); purtroppo, però, se viene inserito dinamicamente tramite *document.write()*, può provocare anche il crash di questo vecchio browser.

Sembra che ciò che faccia "piantare" Netscape 4.x sia l'uso dell'attributo *style="..."* (raccomandato, tra l'altro, dal W3C - vedi [4]) all'interno dei tag generati dinamicamente.

Per questo siamo costretti, nel caso di Netscape 4.x, ad evitare l'uso di tale attri-

buto e usare piuttosto dei tag ed attributi sconsigliati dal W3C o addirittura tag proprietari (per esempio, <LAYER>).

Per Netscape, quindi, dovremo usare il tag <LAYER> (al posto di <DIV>) per definire i *layer* (o livelli), il tag per impostare il tipo di carattere, gli attributi "WIDTH", "HEIGHT", "LEFT", "TOP" per posizionare i *layer*, e così via.

A proposito del tag , si noti la variabile *nav4FontSize* nel file *menu-settings.js*: essa è necessaria in quanto il tag utilizza una scala diversa da quella usata con l'attributo *style="..."*. Poiché per visualizzare i bordi è necessario utilizzare l'attributo *style="..."*, con Netscape 4.x questi non saranno resi.

Il disegno del menu

Come appena detto, il disegno del menu è affidato al metodo *document.write()*; la stringa passata a questo metodo dipende dal browser utilizzato.

Attraverso un ciclo viene scandito l'array *menu* (i cui elementi di indice 0, 2, 4... sono i titoli dei menu), e vengono generati i corrispettivi *layer*; nel caso di un browser diverso da Netscape 4.x la stringa passata al browser sarà del tipo:

```
<div id='menu0' style='position:absolute;
left:20px;right:20px;...'
onmouseover='activate(0)'
onclick='checkForDropDown(0)'
>File</div>
```

Nell'attributo *style* sono contenute, oltre alle informazioni relative al posizionamento, anche quelle relative al font utilizzato, allo spessore del bordo, ed ai vari colori. Nel caso invece di Netscape 4.x la stringa sarà del tipo:

LISTATO 2

Il file "menu-text.js"

```
// File "menu-text.js"
// Contiene le voci che compongono il menu a tendina

menu=[
'Motori di ricerca',
['Altavista','http://www.altavista.com',
'Yahoo','http://www.yahoo.com',
'Google','http://www.google.com',
'Virgilio','http://www.virgilio.it'],
'Posta elettronica',
['Hotmail','http://www.hotmail.com',
'Libero','http://www.libero.it',
'Netaddress','http://www.netaddress.com'],
'Siti vari',
['TeX user group','http://www.tug.org',
'Web Poker','http://www.websamba.com/webpoker',
'Alert','javascript:alert("Prova alert")']
];
```

Il file "menu-code.js"

```

// File "menu-code.js"

var currentMenu=-1;
var dropdown=false;
var justDropped=false;

// Attributi di default, sia per le voci
// principali, sia per i sottomenu (!= Netscape 4.x)
stdAttr="" +
"width:"+rowLength+"px;" +
"border-width:"+borderWidth+"px;" +
"border-style:outset;" +
"border-color:"+borderColor+";" +
"font-family:"+fontName+";" +
"font-size:"+fontSize+"px;" +
"line-height:"+rowHeight+"px;"
if (isN4)
  borderWidth=0.5;
cellHeight=2*borderWidth+rowHeight;

if (!isIE)
  cellWidth=2*borderWidth+rowLength;
else
  cellWidth=rowLength;

// Disegna i menu
for (i=0; i<menu.length/2; i++) {
  menuName=menu[i*2];
  x=startx+(i*cellWidth);
  y=starty;
  if (!isN4)
    str="<div id='menu' + i + '' style='" + stdAttr +
"color:"+mainColor+";" +
"cursor:default;" +
"position:absolute;" +
"left:" + x + "px;" +
"top:" + y + "px;" +
"background-color: "+mainBgColor+";" +
" " +
" onmouseover='activate("+i+")'" +
" onclick='checkForDropDown("+i+")'" +
">" +
  menuName + "</div>";
  else
    str="<layer id='menu' + i + '' left="+x+" top="+y+
" width="+rowLength+" height="+rowHeight+
" bgcolor='"+mainBgColor+"' onmouseover=
'activate("+i+")'>"+
"<a href='javascript:checkForDropDown("+i+")'>"+
"<font face='"+fontName+"' size='"+nav4FontSize+"'
color='"+mainColor+"'>"+
  menuName+
"</font>"+
"</a></layer>\n";
  document.write(str);
  id='menu' + i + "" + j + "" style=
" " + stdAttr +
"visibility:hidden;" +
"position:absolute;" +
"left:" + x + "px;" +
"top:" + y + "px;" +
"background-color: "+itemBgColor+";" +
" " +
" onclick='activate(-1)'" +
">" +
    "<a style='text-decoration:none; width:"+
      (rowLength-2*borderWidth)+"px;" +
      "cursor:default;" +
      "color: "+itemColor+";" +
      "' href='" + itemAction + "'>"+
      itemName+"</a></div>";
  else
    str="<layer id='menu' + i + "" + j + "" left=
      "+x+" top="+y+
      " width="+rowLength+" height="+rowHeight+
      " visibility='hidden'" +
      " bgcolor='"+itemBgColor+"'>"+
      "<a href='"+itemAction+"' onclick='activate(-1)'" +
      "<font face='"+fontName+"' size='"+nav4FontSize+"'
      color='"+itemColor+"'>"+
      itemName+
      "</font>"+
      "</a></layer>\n";
  document.write(str);
}
}

// Gestisce il click all'esterno dei menu
// nel caso il browser non sia Netscape 4.x
if (!isN4) document.onclick = initMenu;

// Visualizza il k-esimo menu (se consentito):
// se k=-1 disattiva tutti i menu
function activate(k) {
  if (!dropdown)
    return;
  justDropped=(currentMenu == -1);
  currentMenu=k;
  for (i=0; i<menu.length/2; i++) {
    items = menu[i*2+1];
    for (j=0; j<items.length/2; j++) {
      if (i==k)
        showLayer(obj("menu"+i+""+j));
      else
        hideLayer(obj("menu"+i+""+j));
    }
  }
  if (k<0)
    dropdown=false;
}

// Abilita/disabilita la visualizzazione dei menu
function checkForDropDown(k) {
  if (!dropdown) {
    dropdown=true;
    activate(k);
  }
  else
    activate(-1);
}

// Disabilita la visualizzazione dei menu
// quando l'utente clicca fuori dei menu
// (non per Netscape 4.x)
function initMenu() {
  if (!justDropped)
    activate(-1);
  else
    justDropped=false;
}

```

```

<layer id='menu0' left=20 top=
      20 width=100 height=15
      bgcolor=#c0c0c0 onmouseover=
      'activate(0) '>
  <a href='javascript:
      checkForDropDown(0) '>
    <font face='verdana' size=
      '2' color='#000000' >
      File</font>
  </a>
</layer>

```

A parte ciò che si è già detto, una differenza che risalta è sulla modalità di chiamata dell'istruzione `checkForDropDown(0)`: nel caso di un browser diverso da Netscape 4.x la chiamata avviene al clic del `layer`, mentre con Netscape 4.x avviene tramite l'attributo `href="..."` di un link `<A>`. Infatti, Netscape non riconosce l'evento `onclick` per un `layer`.

Esaminiamo ora con più cura il significato di questi tag; il `layer` ha come ID "menu0": lo zero indica l'indice del menu (il secondo si chiamerà 'menu1', il terzo 'menu2', ecc.); gli eventi `onmouseover` e `onclick` puntano rispettivamente alle funzioni `activate()` e `checkForDropDown()`, che esamineremo in seguito; per ora diciamo che hanno come parametro l'indice del menu stesso.

Per ogni menu occorre visualizzare i sottomenu: attraverso un ciclo interno a quello principale vengono disegnati anche questi; la stringa passata al browser è analoga a quella appena vista, con alcune differenze. L'ID diventa del tipo 'menu00', 'menu01', 'menu02',... cioè si concatena l'ID della voce principale con l'indice della voce del sottomenu; è presente un attributo "visibility:hidden" (o "visibility=hidden" per Netscape 4.x) per rendere le sottovoci inizialmente invisibili; l'evento `onclick` punta alla funzione "activate(-1)" e le sottovoci sono sempre dei link, che puntano all'URL opportuno indicato dell'array `menu`.

L'attivazione dei menu

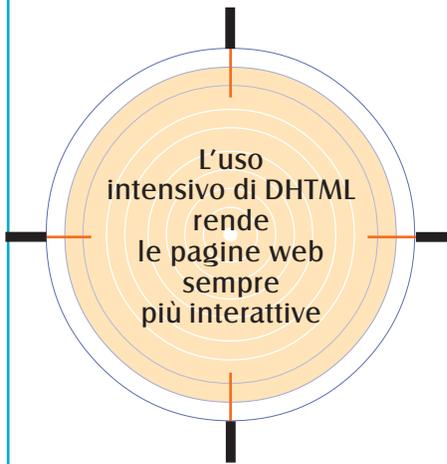
Siamo arrivati al punto più importante: l'attivazione dei menu.

Studiamo innanzitutto la funzione `activate(k)`: essa ha lo scopo di attivare, se consentito (ovvero se `dropdown == true`), il sottomenu `k`-esimo. Prima di ciò, vengono inizializzate due variabili:

- `justDropped`, che varrà `true` se precedentemente non era visualizzato alcun menu, e `false` altrimenti;
- `currentMenu`, che assumerà valore `k` (cioè l'indice del menu correntemente visualizzato).

Successivamente, vengono visualizzate le voci del sottomenu corrente (tramite `showLayer()`, sopra menzionato)

e nascoste le voci che non ne fanno parte (tramite `hideLayer()`). Infine, se il parametro `k` vale `-1` (che equivale a disattivare tutti i menu), la variabile `dropdown` viene impostata a `false`. Ne vedremo tra poco il motivo. L'attivazione dei menu che vogliamo simulare è quella che siamo abituati a vedere nelle finestre di Windows: ad un primo clic viene visualizzato il menu selezionato; da questo momento, se si passa col mouse su di un altro menu (senza cliccare), questo sostituisce il precedente; la "tendina" scompare quando viene fatto clic sul titolo del menu correntemente attivato o (se il browser non è Netscape 4.x) al di fuori dell'area dei menu.



Tutto ciò è controllato dai tre flag di cui ho parlato sopra. Vediamo in dettaglio come funziona il tutto: il flag "dropdown" indica se qualche sottomenu è stato attivato, ed è quindi inizialmente falso. Passiamo col mouse sopra il titolo di un menu (ad esempio sul primo, che ha indice 0): l'evento `onmouseover` punta a `activate(-1)`; questa funzione controlla per prima cosa il valore di verità di `dropdown`: se è falso, esce. Clicchiamo ora su questo menu: l'evento `onclick` punta a `checkForDropDown(0)`, che si occupa di:

1. attivare il menu indicato se `dropdown` è falso;
2. disattivare i menu se `dropdown` è vero.

Quindi, ora il menu viene attivato; inoltre, al passaggio del mouse sul titolo di un altro menu, quest'ultimo verrà attivato in virtù del valore `true` di `dropdown`; tutto termina quando l'utente clicca sul titolo del menu corrente (ancora `checkForDropDown()`).

Resta ora da gestire la disattivazione dei menu al clic su un'area esterna a quella dei menu; a causa di una diversa gestione degli eventi, Netscape 4.x non avrà questa funzionalità (con questo browser bisognerà cliccare sul titolo del menu correntemente attivo per disattivare i

menu); vedremo in seguito questo dettaglio. Si noti prima della definizione della funzione `activate()` l'istruzione:

```
if (!isN4) document.onclick = initMenu;
```

Questa associa al clic sulla pagina la funzione `initMenu()`: a questo punto il clic sul titolo di un menu può causare due eventi: `checkForDropDown()` ma anche `initMenu()`; i browser diversi da Netscape 4.x eseguiranno prima `checkForDropDown()` e poi `initMenu()`, mentre con Netscape 4.x l'ordine di chiamata a volte è imprevedibile (provare per credere!); pertanto, si è preferito evitare la chiamata di `initMenu()` nel caso di Netscape 4.x. Vediamo ora come vanno le cose quando l'utente clicca sul titolo di un menu per attivarlo: non essendoci ancora nessun menu visualizzato, `currentMenu` vale `-1`; viene invocata `checkForDropDown()`, la quale richiama a sua volta `activate()`; questa gestisce proprio il flag `justDropped` e lo pone a `true` se `currentMenu` vale `-1`, false altrimenti; all'uscita di `activate()` (e successivamente di `checkForDropDown()`) viene richiamata `initMenu`, che controlla il valore di `justDropped`: se vale `true`, tale flag viene reimpostato a `false`; altrimenti significa che ho cliccato esternamente all'area dei menu per disattivarli tutti.

Conclusioni

Lo script presentato è abbastanza generale per l'utilizzo in vari siti, ed è una valida alternativa ai classici "bottoni" in formato GIF. Inoltre non fa utilizzo di `java`, `flash` o `plug-in` vari, quindi funziona sul browser "nudo e crudo" (purché non troppo vecchio). È comunque suscettibile di miglioramenti, come ad esempio la gestione di livelli multipli di menu.

Bibliografia

- [1] Francesco Sblendorio, Nicola Tino - *Realizzare il drag'n'drop con DHTML* - Login n. 31
- [2] Dan Livingston, Micah Brown - *CSS e DHTML* - Tecniche Nuove - 2000 - ISBN 88-481-1082-7
- [3] <http://www.mozilla.org> - Sito ufficiale di Mozilla, il browser open source da cui deriva Netscape 6.x
- [4] <http://www.w3.org> - Sito ufficiale del W3C, il World Wide Web Consortium, che fissa standard come HTML, XML, CSS.

Francesco Sblendorio è laureando in Informatica presso l'Università degli Studi di Bari. Si occupa di programmazione in C/C++, Visual Basic e di problematiche legate al Web. Può essere contattato tramite e-mail all'indirizzo sblendorio@infomedia.it