

CSS: guida al layout dei menu di navigazione

I menu di navigazione costituiscono una parte essenziale nel layout di un sito web. Se ben costruiti, aumentano di molto l'esperienza utente, favorendo l'accessibilità e l'usabilità delle pagine. In questo articolo tratteremo i più comuni tipi di menu di navigazione e le varie problematiche incontrate nella loro realizzazione con i CSS.

Scelta degli elementi

Fondamentalmente un menu di navigazione non è altro che un insieme (o più insiemi) di link raggruppati secondo caratteristiche comuni. Poichè i link (elemento a) sono elementi inline, in linea teorica si potrebbe utilizzare qualsiasi contenitore di blocco per raggrupparli (div, p, ecc.).

In realtà la scelta ricade sempre sugli elenchi, e questo per alcune valide ragioni:

- con i CSS disabilitati o non supportati, gli elenchi vengono resi in modo diverso rispetto ai comuni contenitori di blocco. Lynx ad esempio li visualizza con un rientro, ed aggiunge la numerazione per gli elenchi ordinati ed un asterisco come marcatore per quelli non ordinati.
- gli elenchi forniscono un ordine logico e semantico ai link. Per esempio, in un elenco di definizione l'elemento dt può contenere l'etichetta di una sezione, e l'elemento dd un elenco non ordinato di link che fanno riferimento a quella sezione, oppure dt può contenere un elemento a e dd ne fornisce la descrizione.

Tipologie di menu

I menu si dividono in tre grandi categorie: verticali, orizzontali e a discesa. I menu orizzontali in genere vengono posti nella parte superiore o inferiore della pagina (o in entrambe), mentre quelli verticali tendono ad essere inseriti nelle colonne laterali. I menu a discesa possono essere invece sia verticali che orizzontali, e vengono posizionati in base alla tipologia e alle esigenze di layout.

Dato che gli elenchi sono elementi di blocco, la loro formattazione predefinita si adatta particolarmente alla realizzazione di menu verticali. Viceversa, nel caso dei menu orizzontali occorre cambiare la formattazione predefinita degli elenchi al fine di visualizzare le voci sulla stessa riga.

Per quanto riguarda i menu a discesa, per realizzare l'effetto dell'apertura del menu occorre usare una combinazione di proprietà e di selettori che vedremo in seguito.

Menu verticali

Consideriamo questo codice (X)HTML:

```
<ul id="menu">
  <li><a href="#">...</a></li>
  <li><a href="#">...</a></li>
  <li><a href="#">...</a></li>
</ul>
```

(Vedi esempio)

Dall'esempio si nota come i link vengano posti uno sopra l'altro. Dato che l'elenco non ordinato segue la sua formattazione predefinita, ha un rientro ed un marcatore per ogni voce. Possiamo cominciare con l'eliminare il marcatore e il rientro, gestendo al contempo lo spazio intorno al menu, ed assegnando a quest'ultimo una dimensione:

```
#menu {  
list-style: none;  
margin: 1em 0;  
padding: 0;  
width: 15em;  
}
```

(Vedi esempio)

Dare una dimensione al menu (e come vedremo anche alle voci e ai link al suo interno) serve a prevenire alcuni problemi comuni in Internet Explorer Windows. Passiamo quindi a gestire le voci del menu:

```
#menu li {  
margin: 0 0 0.2em 0;  
padding: 0;  
width: 15em;  
}
```

(Vedi esempio)

Abbiamo semplicemente distanziato le voci in verticale di 0.2em. A questo punto, per ottenere un tipico menu a blocchi, è necessario trasformare i link in elementi di blocco ed assegnare loro degli stili:

```
#menu a:link,  
#menu a:visited {  
display: block;  
margin: 15px;  
padding: 0.2em 0 0.2em 0.2em;
```

```
color: #c00;  
background: transparent;  
text-decoration: none;  
font-weight: bold;  
}  
#menu a:hover {background: #c00; color: #fff;}
```

(Vedi esempio)

Abbiamo usato la proprietà 'display' impostandola su 'block' per l'elemento a, a cui abbiamo assegnato una dimensione al fine di evitare malfunzionamenti in Internet Explorer Windows. Quindi abbiamo eliminato la sottolineatura dei link tramite la proprietà 'text-decoration', e successivamente abbiamo creato un effetto di rollover cambiando il colore del testo e lo sfondo sullo stato :hover.

Volendo aggiungere dei bordi al menu e alle voci, il codice sinora esaminato andrebbe modificato come segue:

```
#menu {  
list-style: none;  
margin: 1em 0;  
padding: 0;  
width: 15em;  
border: 1px solid #ccc;  
border-bottom: none;  
}  
  
#menu li {  
margin: 0;  
padding: 0;  
width: 15em;
```

```
border-bottom: 1px solid #ccc;
}
```

(Vedi esempio)

Il problema nasce quando vogliamo allineare il testo a sinistra e contemporaneamente aggiungere del padding per distanziarlo dai bordi. Poichè i link sono larghi quanto il blocco contenitore, questo causerà un allargamento del box, più evidente se abbiamo impostato un rollover. In questo caso è sufficiente impostare un padding per il blocco contenitore del menu, al fine di contenerne l'allargamento. Il padding dovrà essere uguale o superiore al padding impostato per i link.

Abbiamo sinora usato il padding per gestire lo spazio verticale del testo, ma è possibile usare anche un'altra soluzione:

```
#menu a:link,
#menu a:visited {
display: block;
width: 15em;
margin: 0;
padding: 0;
color: #c00;
background: transparent;
text-decoration: none;
font-weight: bold;
text-align: center;
height: 1.8em;
line-height: 1.8em;
}
#menu a:hover {background: #c00; color: #fff;}
```

(Vedi esempio)

La centratura verticale del testo in un blocco si ottiene mediante due valori uguali per le proprietà 'height' e 'line-height'. Il valore dell'altezza deve essere uguale al valore dell'interlinea, in modo da avere un eguale spazio sopra e sotto il testo. Dato che abbiamo reso i link come elementi di blocco, è possibile usare la proprietà 'text-align' per l'allineamento orizzontale del testo.

Menu verticali avanzati

Abbiamo visto come la formattazione di base di un menu verticale sia alquanto semplice da realizzare, fermo restando il fatto che vanno rispettate alcune regole di compatibilità con Internet Explorer Windows:

- il menu e le sue voci devono avere una dimensione dichiarata. Anche se, di norma, la dimensione non dichiarata di un elemento di blocco viene calcolata automaticamente partendo dalla dimensione di un antenato, questo non vale in Explorer, che aggiungerà dello spazio verticale tra le voci e non estenderà l'area cliccabile del link a tutto il blocco.
- si eviti di esprimere il padding in percentuale. Valori di questo tipo possono causare in Explorer il cosiddetto *jump on :hover*, ossia il riflusso degli elementi della pagina.

Detto questo, uno delle caratteristiche certamente più richieste dai designer è il rollover di immagini. Questo anticamente veniva realizzato usando JavaScript, ma oggi si può ottenere con i CSS:

```
#menu {  
margin: 0;  
padding: 0;  
width: 15em;  
}
```

```
#menu li {
margin: 0.5em 0;
padding: 0;
list-style: none;
}

#menu a:link,
#menu a:visited {
margin: 0;
padding: 0.35em;
text-decoration: none;
border: 1px solid #000;
background: transparent;
color: #c00;
display: block;
width: 15em;
font-weight: bold;
}
#menu a:hover {
background: #c00 url("img/dar.gif") no-repeat 95%
50%;
color: #fff;
}

#menu strong.active {
border: 1px solid #000;
font-weight: bold;
background: url("img/dar.gif") no-repeat 95% 50%;
color: #000;
padding: 0.35em;
text-transform: uppercase;
width: 15em;
display: block;
}
```

(Vedi esempio)

Anche questa tecnica è relativamente semplice: si tratta di impostare un'immagine di sfondo sullo stato :hover del link usando la proprietà 'background'. Il posizionamento dell'immagine deve essere tale da porla all'interno del blocco contenitore del link. In questo esempio abbiamo valori percentuali per la proprietà 'background-position', perchè garantiscono una certa flessibilità. Volendo si può realizzare con i CSS anche un preload delle immagini, al fine di velocizzare il rollover:

```
#menu a:link,  
#menu a:visited {  
background: transparent url("img/dar.gif") no-repeat  
-200% 0;  
}
```

In questo caso abbiamo impostato un valore negativo per la proprietà 'background-position', in modo che l'immagine non sia visibile sullo stato normale del link ma venga comunque caricata nella cache del browser. Si raccomanda di usare valori negativi, in quanto Safari ha dei problemi con valori positivi nel preload. Se infatti avessimo scritto:

```
#menu a:link,  
#menu a:visited {  
background: transparent url("img/dar.gif") no-repeat  
16em 0;  
}
```

gli altri browser non avrebbero mostrato l'immagine, in quanto si trova fuori dal blocco contenitore dell'elemento, ma Safari l'avrebbe comunque resa visibile fuori dal elemento a.

Si noti come nell'esempio abbiamo specificato un *indicatore di sezione*, ossia una voce del menu formattata in modo diverso dalle altre per ricordare all'utente la sua posizione all'interno del sito. Abbiamo usato un elemento `strong` e non un elemento `a`, in quanto con i CSS disabilitati l'elemento `a` si confonderebbe con gli altri link, disorientando l'utente:

```
#menu strong.active {
border: 1px solid #000;
font-weight: bold;
background: url("img/dar.gif") no-repeat 95% 50%;
color: #000;
padding: 0.35em;
text-transform: uppercase;
width: 15em;
display: block;
}
```

Infine, se avessimo voluto scambiare le immagini al passaggio del mouse avremmo solo dovuto aggiungere le seguenti dichiarazioni:

```
#menu a {background: transparent url("img/dar.gif")
no-repeat -200% 0;}
/* preload */

#menu a:link,
#menu a:visited {background-position: 95% 50%;}

#menu a:hover {background-image:
url("img/dar2.gif");}
```

Il preload con i CSS è utile quando le immagini pesano da 1Kb in su, o quando vi sono più menu con diverse immagini di sfondo o più immagini di sfondo per lo stesso menu. In caso contrario si può anche omettere la procedura descritta sopra.

È possibile anche usare i bordi in modo dinamico, come mostrato in questo esempio:

```
#menu a:link,
#menu a:visited {
display: block;
width: 15em;
margin: 0;
padding: 0.2em 0 0.2em 0.2em;
color: #c00;
background: transparent;
text-decoration: none;
font-weight: bold;
border-right: 5px solid #fff;
border-bottom: 1px solid #fff;
}
#menu a:hover {
color: #000;
background: transparent;
border-right-color: #ccc;
border-bottom-color: #ccc;
}
```

(Vedi esempio)

Abbiamo impostato un colore di partenza del bordo uguale a quello dello sfondo del menu (ed in questo caso anche della finestra di visualizzazione)

per poi cambiarlo sullo stato : hover , creando l'effetto di un cursore che si sposta e di una sottolineatura per il blocco.

Menu orizzontali

Le tecniche sinora esposte si applicano anche ai menu orizzontali, ma con la differenza che nel caso di tali menu occorre cambiare le regole di visualizzazione per le voci. Se per i menu verticali non vi era bisogno di specificare una formattazione particolare per le voci, dato che gli elenchi si estendono per loro stessa natura in verticale, nel caso dei menu orizzontali occorre agire sulle voci usando la proprietà 'float' o la dichiarazione 'display: inline', a seconda delle esigenze e della tipologia di layout.

Vediamo un semplice esempio:

```
#menu {
list-style: none;
margin: 1em 0;
padding: 0;
border-bottom: 1px solid #000;
}

#menu li {
margin: 0;
padding: 0 0.5em;
display: inline;
border-right: 1px solid #000;
height: 0;
}

#menu a:link,
#menu a:visited {
margin: 0;
```

```
padding: 0;
color: #c00;
background: transparent;
text-decoration: none;
font-weight: bold;
}
#menu a:hover {
color: #000;
background: transparent;
}

#menu li.last {border-right: none;}
```

(Vedi esempio)

La dichiarazione 'display: inline' per l'elemento `li` fa in modo che le voci vengano visualizzate affiancate. La dichiarazione 'height: 0' serve unicamente ad Internet Explorer 5 Windows, che di norma non riconosce margini, padding e bordi agli elementi inline a meno che questi non "abbiano layout" (secondo la terminologia inerente alla proprietà `hasLayout` di Explorer). Con questa dichiarazione Explorer 5 considera tali elementi come ibridi di blocco e inline, applicando loro le sopracitate proprietà. Tuttavia, i risultati non sono perfetti. La classe `.last` elimina il bordo dall'ultima voce del menu, creando una simmetria con la prima. La misura tra i link è di 0.5em, ossia il valore universalmente accettato nelle pratiche di accessibilità.

La soluzione con 'display: inline' viene usata soprattutto per la facilità con cui si possono allineare le voci del menu. Infatti, trattandosi di elementi inline è sufficiente usare la proprietà 'text-align'. Purtroppo i problemi di compatibilità con Explorer 5 Windows sono così gravi che per ottenere un risultato compatibile spesso si è costretti ad aggiungere una notevole quantità di codice solo per Explorer.

Usare la proprietà 'float' è invece meno problematico. Il sopracitato esempio potrebbe essere così riscritto:

```
#menu {  
list-style: none;  
margin: 1em 0;  
padding: 0;  
border-bottom: 1px solid #000;  
height: 1.5em;  
line-height: 1.5em;  
}
```

```
#menu li {  
float: left;  
width: 3em;  
margin: 0 0.5em 0 0;  
padding: 0;  
border-right: 1px solid #000;  
}
```

```
#menu a:link,  
#menu a:visited {  
margin: 0;  
padding: 0;  
color: #c00;  
background: transparent;  
text-decoration: none;  
font-weight: bold;  
}  
#menu a:hover {  
color: #000;  
background: transparent;  
}
```

```
#menu li.last {border-right: none;}
```

(Vedi esempio)

Con il float dobbiamo necessariamente assegnare un'altezza al menu, o lo spazio verticale risulterebbe collassato. In questo caso abbiamo assegnato un valore di 'line-height' pari all'altezza per centrare il testo in verticale. Le voci di menu devono avere una larghezza esplicita, che deve essere proporzionale al contenuto testuale delle medesime. Lo svantaggio maggiore di questa tecnica consiste, a detta di molti, nella difficoltà di allineamento delle voci. Per esempio, per centrare le voci all'interno del menu dovremmo aggiungere il seguente codice:

```
#contenitore-menu {  
margin: 1em 0;  
text-align: center;  
border-bottom: 1px solid #000;  
}
```

```
#menu {  
list-style: none;  
margin: 0 auto;  
padding: 0;  
height: 1.5em;  
line-height: 1.5em;  
width: 19%;  
}
```

```
#menu li {  
float: left;  
width: 3em;
```

```
margin: 0 0.5em 0 0;
padding: 0;
border-right: 1px solid #000;
}

#menu a:link,
#menu a:visited {
margin: 0;
padding: 0;
color: #c00;
background: transparent;
text-decoration: none;
font-weight: bold;
}
#menu a:hover {
color: #000;
background: transparent;
}

#menu li.last {border-right: none;}
```

(Vedi esempio)

Abbiamo inserito il menu all'interno di un contenitore (`#contenitore-menu`), a cui abbiamo assegnato la dichiarazione `'text-align: center'` per centrare il menu in Explorer 5 Windows. Quindi abbiamo assegnato al menu una larghezza esplicita (19%), che è ovviamente proporzionale al numero di voci del menu. Per centrare il menu sugli altri browser abbiamo impostato i suoi margini orizzontali su `'auto'`.

Nel caso in cui dovessimo allineare le voci su di un lato, il precedente codice andrebbe modificato come segue:

```
#contenitore-menu {  
margin: 1em 0;  
border-bottom: 1px solid #000;  
height: 1.5em;  
}
```

```
#menu {  
list-style: none;  
margin: 0;  
padding: 0;  
height: 1.5em;  
line-height: 1.5em;  
width: 19%;  
float: right;  
}
```

```
#menu li {  
float: left;  
width: 3em;  
margin: 0 0.5em 0 0;  
padding: 0;  
border-right: 1px solid #000;  
}
```

```
#menu a:link,  
#menu a:visited {  
margin: 0;  
padding: 0;  
color: #c00;  
background: transparent;  
text-decoration: none;  
font-weight: bold;
```

```
}  
#menu a:hover {  
  color: #000;  
  background: transparent;  
}  
  
#menu li.last {border-right: none;}
```

(Vedi esempio)

In questo caso abbiamo fatto flottare il menu a destra e le sue voci a sinistra, avendo l'accortezza di impostare un'altezza per `#contenitore-menu`.

Menu orizzontali avanzati

Per i menu orizzontali avanzati vale quanto abbiamo visto per i menu verticali avanzati. Ad esempio, è possibile creare un rollover di immagini al passaggio del mouse:

```
#menu {  
  margin: 0;  
  padding: 0;  
  height: 2em;  
}  
  
#menu li {  
  margin: 0 0.5em 0 0;  
  padding: 0;  
  list-style: none;  
  height: 2em;  
  float: left;  
}
```

```
#menu a:link,
#menu a:visited {
margin: 0;
padding: 0.35em;
text-decoration: none;
border: 1px solid #000;
background: transparent;
color: #c00;
display: block;
width: 10em;
float: left;
font-weight: bold;
}
#menu a:hover {
background: #c00 url("img/dar.gif") no-repeat 98%
50%;
color: #fff;
}

#menu strong.active {
border: 1px solid #000;
font-weight: bold;
background: url("img/dar.gif") no-repeat 98% 50%;
color: #000;
padding: 0.35em;
text-transform: uppercase;
float: left;
width: 10em;
margin: 0;
}
```

(Vedi esempio)

In questo caso anche l'elemento `a` è stato fatto flottare, al fine di trasformarlo in un elemento di blocco. Si noti come l'elemento `a` e l'elemento `li` siano dimensionati, avendo il primo una larghezza ed il secondo un'altezza (pari all'altezza del menu).

Si possono creare anche effetti di movimento, come nel seguente esempio:

```
#menu {
margin: 0;
padding: 0;
height: 3em;
}

#menu ul {
list-style: none;
margin: 0;
padding: 0;
height: 3em;
}

#menu li {
float: left;
padding: 0;
margin: 0 0.5em;
display: inline;
}

#menu a:link,
#menu a:visited {
margin: 0;
padding-left: 16px;
width: 4em;
line-height: 1em;
float: left;
```

```
color: #c00;
background: transparent url("img/ar.gif") no-repeat 0
50%;
font-weight: bold;
text-align: left;
text-decoration: none;
}
#menu a:hover {
line-height: 2em;
color: #000;
background: transparent url("img/check2.gif") no-
repeat 0 50%;
}
```

(Vedi esempio)

In questo caso abbiamo aumentato l'interlinea al passaggio del mouse al fine di creare l'abbassamento dinamico delle voci del menu. Da notare come l'elemento `li`, che ha il margine sinistro con valore `0.5em`, abbia la dichiarazione `'display: inline'`, il cui unico scopo è quello di evitare il raddoppiamento del suddetto margine in Internet Explorer Windows.

Un altro effetto che possiamo creare è quello di un posizionamento verticale alternato delle voci del menu, come in questo esempio:

```
#menu {
margin: 0;
padding: 0;
height: 2em;
}

#menu li {
margin: 0 0.5em 0 0;
```

```
padding: 0;
list-style: none;
height: 2em;
float: left;
}
```

```
#menu li.basso {margin-top: 1em;}
```

```
#menu a:link,
#menu a:visited {
margin: 0;
padding: 0.35em;
text-decoration: none;
border-width: 1px;
border-style: solid;
background: transparent;
color: #c00;
display: block;
width: 10em;
float: left;
font-weight: bold;
}
#menu a:hover {
background: #c00 url("img/dar.gif") no-repeat 98%
50%;
color: #fff;
}
```

```
#menu strong.active {
border: 1px solid #000;
font-weight: bold;
background: transparent url("img/dar.gif") no-repeat
98% 50%;
color: #000;
```

```
padding: 0.35em;
text-transform: uppercase;
float: left;
margin: 0;
width: 10em;
}
```

(Vedi esempio)

In questo caso abbiamo creato la classe `.basso` che viene applicata in modo alternato alle voci di menu in modo da spingerle verso il basso tramite un margine superiore pari a 1em.

Infine, un effetto molto ricercato è quello di un menu con linguette (o *tabs*), come mostrato nel seguente esempio:

```
#menu {
margin: 0;
padding: 0;
height: 2em;
border-bottom: 1px solid #000;
}
```

```
#menu li {
margin: 0 0.5em 0 0;
padding: 0;
float: left;
height: 2em;
width: 5em;
list-style-type: none;
}
```

```
#menu li a:link,
```

```
#menu li a:visited {
margin: 0;
padding: 0;
text-decoration: none;
border-width: 1px 1px 0 1px;
border-style: solid;
color: #c00;
background: #fff;
height: 2em;
line-height: 2em;
float: left;
width: 5em;
text-align: center;
font-weight: bold;
position: relative; /* per Internet Explorer */
}
#menu li a:hover {
background: #c00; color: #fff;
}

#menu li strong {
float: left;
width: 5em;
height: 2em;
background: #fff;
color: #000;
padding: 0;
line-height: 2em;
text-align: center;
text-transform: uppercase;
}
```

Il codice aggiuntivo per Internet Explorer Windows:

```
<!--[if lt IE 7]>
<style type="text/css">
#menu li {margin-bottom: -1px;}
</style>
<![endif]-->
```

(Vedi esempio)

Fondamentalmente, nella realizzazione di un menu a tabs occorre stabilire il corretto allineamento verticale delle voci del menu, spingendole a ridosso del bordo inferiore del menu fino a quando lo sfondo delle voci non copre tale bordo. Ovviamente le voci non devono avere un bordo inferiore.

Menu a discesa

I menu a discesa non sono altro che menu orizzontali o verticali che contengono a loro volta degli altri menu, ossia degli elenchi che contengono altri elenchi annidati al loro interno. Il codice (X)HTML visto sinora va quindi riscritto come segue:

```
<ul id="menu">
    <li><a href="#">...</a>
        <ul>
            <li><a href="#">...</a></li>
        </ul>
    </li>
    ...
</ul>
```

Nei menu a discesa il dinamismo di apertura viene creato dalla pseudo-classe dinamica `:hover` applicata all'elemento `li`. Per nascondere e per

mostrare il contenuto del sotto-menu si fa ricorso al posizionamento contestuale unito all'alternanza tra 'display: none' e 'display: block'. Vediamo un esempio:

```
#menu {  
margin: 0;  
padding: 0;  
height: 2em;  
border-width: 0 0 1px 0;  
border-style: solid;  
}
```

```
#menu li {  
margin: 0;  
padding: 0;  
float: left;  
height: 2em;  
width: 10em;  
display: inline;  
list-style-type: none;  
position: relative;  
line-height: 2em;  
}
```

```
#menu li a:link,  
#menu li a:visited {  
margin: 0;  
padding: 0;  
text-decoration: none;  
color: #000;  
background: transparent;  
float: left;  
width: 10em;
```

```
text-align: center;
font-weight: bold;
}
#menu li a:hover {
background: transparent;
color: #c00;
}

#menu li ul {
display: none;
margin: 0;
padding: 0;
background: #f1f1f1;
color: #000;
border: 1px solid;
width: 10em;
}
#menu li ul li {
list-style: none;
margin-left: 0;
padding-left: 0;
border-bottom: 1px solid;
}

#menu li ul li a:link,
#menu li ul li a:hover {
display: block;
margin: 0;
padding: 0;
text-align: center;
background: transparent;
color: #000;
width: 10em;
font-weight: bold;
```

```
}  
#menu li ul li a:hover {  
background: #fff;  
color: #c00;  
}
```

```
#menu li:hover ul {  
display: block;  
position: absolute;  
top: 2em;  
left: 0;  
}
```

(Vedi esempio)

Viene creato un posizionamento contestuale ('position: relative') sull'elemento `li` del menu principale. Quindi il sotto-menu all'interno dell'elemento `li` viene nascosto e formattato con le seguenti dichiarazioni:

```
#menu li ul {  
display: none;  
margin: 0;  
padding: 0;  
background: #f1f1f1;  
color: #000;  
border: 1px solid;  
width: 10em;  
}
```

A questo punto, sfruttando il posizionamento contestuale appena creato e la pseudo-classe dinamica `:hover` sull'elemento `li` del menu principale, si

fa comparire il sotto-menu dichiarandolo di blocco e posizionandolo in modo assoluto rispetto all'elemento `li` del menu principale:

```
#menu li:hover ul {  
display: block;  
position: absolute;  
top: 2em;  
left: 0;  
}
```

Poichè Internet Explorer 6 (ed inferiori) non supporta la pseudo-classe dinamica `:hover` sugli elementi diversi dai link, occorre dare a questo browser uno stile diverso per permettere agli utenti di poter visualizzare il sotto-menu:

```
<!--[if lt IE 7]>  
<style type="text/css">  
#menu li ul {display: block;}  
</style>  
<![endif]-->
```

Si possono creare menu che compaiono passando con il mouse sopra un'immagine, come in questo esempio:

```
#menu {  
margin: 0;  
padding: 0;  
width: 22px;  
position: absolute;  
top: 20%;
```

```
left: 0;
background: #f1f1f1;
color: #000;
border-width: 1px;
border-style: solid;
}
```

```
#menu ul {
margin: 0.5em 0 0.5em 2.5em;
padding: 0;
list-style: none;
display: none;
}
```

```
#menu ul li a:link,
#menu ul li a:visited {
display: block;
margin: 0;
padding: 0.2em;
color: #329;
text-decoration: none;
width: 5em;
}
#menu ul li a:hover {
background: #fff url("img/dar.gif") no-repeat 100%
50%;
color: #000;
}
```

```
#menu p {margin: 0; text-align: right; padding: 0;}
#menu p img {border: none; vertical-align: middle;}
```

```
#menu:hover {width: 10em;}
#menu:hover ul {display: block;}
```

(Vedi esempio)

In questo caso, come nel precedente, il sotto-menu viene prima nascosto tramite 'display: none' e quindi rivelato su :hover tramite la dichiarazione 'display: block'. Ancora una volta è necessario dare ad Explorer 6 (ed inferiori) del codice a parte:

```
<!--[if lt IE 7]>
<style type="text/css">
#menu ul {display: block;}
</style>
<![endif]-->
```

Infine, è possibile combinare dinamicamente i menu orizzontali con quelli verticali, come mostrato da questo esempio:

```
#menu {
margin: 0;
padding: 0;
height: 2em;
border-width: 0 0 1px 0;
border-style: solid;
}

#menu li {
margin: 0;
padding: 0;
float: left;
height: 2em;
width: 10em;
display: inline;
list-style-type: none;
```

```
position: relative;
line-height: 2em;
}
```

```
#menu li a:link,
#menu li a:visited {
margin: 0;
padding: 0;
text-decoration: none;
color: #000;
background: transparent;
float: left;
width: 10em;
text-align: center;
font-weight: bold;
}
#menu li a:hover {
background: transparent;
color: #c00;
}
```

```
#menu li ul,
#menu li ul li.apri ul {
display: none;
margin: 0;
padding: 0;
background: #f1f1f1;
color: #000;
border: 1px solid;
width: 10em;
}
#menu li ul li {
list-style: none;
margin-left: 0;
```

```
padding-left: 0;
border-bottom: 1px solid;
}

#menu li ul li a:link,
#menu li ul li a:hover {
display: block;
margin: 0;
padding: 0;
text-align: center;
background: transparent;
color: #000;
width: 10em;
font-weight: bold;
}
#menu li ul li a:hover {
background: #fff;
color: #c00;
}

#menu li:hover ul {
display: block;
position: absolute;
top: 2em;
left: 0;
}
#menu li.apri:hover ul {
display: block;
position: absolute;
top: 0;
left: 10em;
}
#menu li ul li.apri a.link {
background-image: url("img/dar.gif");
```

```
background-repeat: no-repeat;
background-position: 98% 50%;
}
```

(Vedi esempio)

In questo esempio abbiamo usato le stesse tecniche viste in precedenza, con la differenza che è stata creata una classe `.apri` per l'elemento `li` del menu principale, al fine di visualizzare il sotto-menu solo su una determinata voce. Per indicare che tale voce contiene un sotto-menu, abbiamo usato un'immagine di sfondo sul link contenuto in essa. Internet Explorer 6 (ed inferiori) necessita del seguente codice aggiuntivo:

```
<!--[if lt IE 7]>
<style type="text/css">
#menu li ul {display: block;}
#menu li ul li.apri ul {display: block;}
</style>
<![endif]-->
```

Nota sui behavior HTC

I menu a discesa visti in precedenza si sarebbero potuti rendere compatibili anche con Internet Explorer 6 (ed inferiori) tramite l'uso dei file `.htc` usando la seguente dichiarazione data tramite commenti condizionali:

```
body {behavior: url(csshover.htc);}
```

Purtroppo i file HTC hanno le seguenti controindicazioni:

- Se il Service Pack 2 è installato su Windows, il file `.htc` funzionerà in Internet Explorer 6 solo se servito col il MIME type `text/x-component`. È necessario contattare l'amministratore del server per le eventuali modifiche.
- Internet Explorer avrà problemi se il suo livello di sicurezza è impostato su *Alto*. Per questo si raccomanda l'uso dell'elemento `noscript` all'interno del commento condizionale.

Ovviamente l'ultima raccomandazione si applica anche nel caso in cui l'utente abbia disabilitato gli script sul suo browser. Potete comunque scaricare il file `csshover.htc` in formato compresso qui.

Accessibilità dei menu di navigazione

Oltre all'uso delle accesskey, per garantire una buona accessibilità dei menu di navigazione si dovrebbe evitare di usare la dichiarazione `list-style: none` per gli elenchi non ordinati e ordinati e le voci di elenco, in quanto tale dichiarazione impedisce ai lettori di schermo di stabilire il corretto ordinamento delle voci. Per eliminare il marcatore di elenco, si usi invece `display: block` o `display: inline` per l'elemento `li`.

Ringrazio Michele Diodati per la preziosa segnalazione.