



# Drop-Down menu vertical - Parte 1

Autor: [Nick Rigby](#)

URL do original: <http://www.alistapart.com/articles/horizdropdowns/>

Título original: Drop-Down Menus, Horizontal Style

Translated with the permission of [A List Apart Magazine](#) and the author.

Permissions and copyrights at: <http://www.alistapart.com/copyright/>

## Introdução

Qualquer um que tenha projetado um menu drop-down estará familiarizado com a grande quantidade de scripts necessários para fazer funcionar estes tipos de menus. Mas, usando-se uma marcação HTML estruturada e regras CSS simples é possível criar menus drop-down de raro apelo visual, fáceis de editar e atualizar e que funcionem perfeitamente em vários tipos de browsers, inclusive no Internet Explorer. Melhor ainda, para designers zelosos de seus códigos, nenhum JavaScript é necessário! (Na verdade, um 'pouquinho' de JavaScript é necessário, mas nada tão complicado, como você deve estar imaginando.)

Aqui temos uma visão antecipada do menu em ação.

## Criando o menu

O primeiro e mais importante passo na criação do menu é a definição de sua estrutura. A melhor maneira de se criar a estrutura é através de uma lista não ordenada, com cada um dos sub-menus constituídos por outra lista não ordenada dentro da primeira. Parece complicado? É bem direto, observe:

```
<ul>
  <li><a href="#">Home</a></li>
  <li><a href="#">About</a>
    <ul>
      <li><a href="#">History</a></li>
      <li><a href="#">Team</a></li>
      <li><a href="#">Offices</a></li>
    </ul>
  </li>
  <li><a href="#">Services</a>
    <ul>
      <li><a href="#">Web Design</a></li>
      <li><a href="#">Internet
        Marketing</a></li>
      <li><a href="#">Hosting</a></li>
      <li><a href="#">Domain Names</a></li>
      <li><a href="#">Broadband</a></li>
    </ul>
  </li>
  <li><a href="#">Contact Us</a>
    <ul>
```

```
<li><a href="#">United Kingdom</a></li>
<li><a href="#">France</a></li>
<li><a href="#">USA</a></li>
<li><a href="#">Australia</a></li>
</ul>
</li>
</ul>
```

É isso aí: um HTML simples, ao mesmo tempo acessível e fácil de editar.

## Visualmente atraente?

Se você visitou a página contendo a visão antecipada do menu (link lá em cima), visualizou uma entediante lista de itens. E, eu prometo iremos torná-la mais atrativa! Vamos adicionar-lhe estilos.

Primeiramente removeremos as marcas (bullets) e a indentação próprias das listas não ordenadas e definiremos uma largura para os itens do menu.

```
ul {
    margin: 0;
    padding: 0;
    list-style: none;
    width: 150px;
}
```

A seguir precisamos posicionar os itens da lista. Felizmente os itens são renderizados um embaixo do outro, por padrão, que é justamente o que nós queremos. Contudo vamos definir o posicionamento com o valor `relative`, porque iremos posicionar os sub-menus de modo absoluto em relação a eles, itens.

```
ul li {
    position: relative;
}
```

Próximo passo, são os sub-menus. Queremos cada sub-menu ao lado direito do seu elemento pai (item) quando passarmos o mouse sobre ele, item.

```
li ul {
    position: absolute;
    left: 149px;
    top: 0;
    display: none;
}
```

Usando as propriedades CSS “left” e “top” posicionamos de modo absoluto cada sub-menu ao lado de seu item pai. Você deve ter notado que eu defini para a propriedade “left” o valor de 149px (1px menor

que a largura dos itens do menu), isto faz com que os sub-menus transpassem o menu e em consequência não apareça uma borda dupla. (Você verá este efeito adiante.)

Também definimos display como “none,” pois não queremos os sub-menus visíveis por padrão.

Bem, agora já temos nosso framework no lugar, mas isto ainda está com um visual bem precário. Vamos estilizar nossos links.

```
ul li a {
    display: block;
    text-decoration: none;
    color: #777;
    background: #fff;
    padding: 5px;
    border: 1px solid #ccc;
    border-bottom: 0;
}
```

Eu estiliziei os links de acordo com minhas preferências pessoais, mas é claro, você pode estilizar segundo as suas necessidades. É importante definir display como “block,” para garantirmos que cada link ocupará toda a largura definida para o item da lista.

Bem, as coisas agora parecem um pouco melhor, muito embora usuários com Internet Explorer não concordem. Lamentavelmente o IE Win interpreta as quebras de linha entre os itens de lista, como espaços vazios e assim para o IE os itens do menu não se assentam um sobre o outro. Contudo existe uma maneira de corrigir este bug do IE:

```
/* Fix IE. Hide from IE Mac */
* html ul li { float: left; }
* html ul li a { height: 1%; }
/* End */
```

Podemos aplicar o Holly Hack acima, que esconde estas regras CSS de todos os browsers menos do IE Win. Perfeito. Você deve ter notado a declaração `height: 1%` que foi definida. Lamentavelmente (outra vez!) a propriedade float usada no hack faz aparecer outro bug no IE, que requer a declaração de um valor para height a fim de fazer os links elementos nível de bloco.

Você deve ter notado também que se faz necessário 'fechar' o menu, e isto pode ser feito adicionando-se uma borda no fim da lista. E, as regras CSS para ul são:

```
ul {
    margin: 0;
    padding: 0;
    list-style: none;
    width: 150px;
    border-bottom: 1px solid #ccc;
```

```
}
```

Com alguma dose de sorte, todos agora estaremos aptos a visualizar o menu não funcionando.

## Fazendo funcionar

Agora o pulo! Precisamos fazer 'aparecer' os sub-menus quando passamos o mouse sobre o item.

```
li:hover ul { display: block; }
```

Oba...aqui nosso menu em ação.

“Woo hoo! Funciona!” ouço 1% de vocês exclamar. “Sinistro!”

OK, OK, droga, o IE/Win tinha que arruinar tudo e não se comportar em conformidade. O IE/Win somente suporta a pseudo classe `:hover` aplicada a um link — assim `li:hover` que faz os sub-menus aparecer, nada significa para o IE.

Um 'pouquinho' de JavaScript será necessário para trazer o IE à ação. (line wraps marked » — *Ed.*):

```
startList = function() {  
  if (document.all&&document.getElementById) {  
    navRoot = document.getElementById("nav");  
    for (i=0; i<navRoot.childNodes.length; i++) {  
      node = navRoot.childNodes[i];  
      if (node.nodeName=="LI") {  
        node.onmouseover=function() {  
          this.className+=" over";  
        }  
        node.onmouseout=function() {  
          this.className=this.className.replace  
            (" over", "");  
        }  
      }  
    }  
  }  
}  
window.onload=startList;
```

Muito grato e apreciado à Patrick Griffiths e Dan Webb, que mostraram esta dica em um artigo no Suckerfish Dropdowns. Obrigado, rapazes!

Então, a regra para hover agora é:

```
li:hover ul, li.over ul {  
  display: block;  
}
```

Adicionalmente precisamos referenciar o JavaScript com nosso `ul`, principal, assim:

```
<ul id="nav">
```

Esperamos então, agora com as coisas nos seus lugares, que todos sejamos capazes de visualizar uma versão simples do menu em ação.

## IE5.01 Jumping Menu Bug

Quem estiver usando IE5.01 em Windows notará que o menu tende a 'saltar' quando passamos o mouse sobre determinados itens. Este problema é facilmente resolvido modificando nosso hack como mostrado a seguir:

```
/* Fix IE. Hide from IE Mac */  
* html ul li { float: left; height: 1%; }  
* html ul li a { height: 1%; }  
/* End */
```

## Um misterioso bug no IE6 :

Durante o desenvolvimento deste artigo, eu descobri um estranho bug, que acredito acontecer apenas no IE6. Devemos declarar um background para `li a`, pois quando um sub-menu estende-se verticalmente além do menu, os links desaparecem antes que se tenha tempo de clicá-los. Estranho! Tente você mesmo.

## Faça você mesmo

É isso aí! Uma maneira standard para criar um interessante drop-down menu horizontal. Tudo o que você tem a fazer agora é personalizar a estilização e criar seu próprio menu. Para servir de 'aperitivo' [um menu com mais apelo visual](#). Eu o preparei antes. Divirta-se!

*NT: O autor publicou dois artigos posteriores, mostrando como acrescentar mais camadas de sub-sub-menus e incrementando o menu atual.*

*Traduzi tais artigos em 2005-9-19 e você poderá continuar lendo a [Parte 2](#)*