

分享文档系列

TreeView 用户指南

作者： 分享文档
日期：2007-3-22
版本：V1.1(2007-03-22)
较对：hunqiu

本文是分享文档站长胡萝卜的作品。大家可以免费阅读、在网络上进行分发，前提是必须保留本文档的完整性。

分享文档是一家专业的 Java 技术网站，给中国 Java 程序员提供各种 Java 资源如文档，工具，教程，社区交流等。

我们的官方网站是：<http://chinesedocument.com>

我们的官方论坛是：<http://bbs.chinesedocument.com>

请大家多多关注分享文档，我们还会发布更多优秀的文档！

前 言

什么也不写。

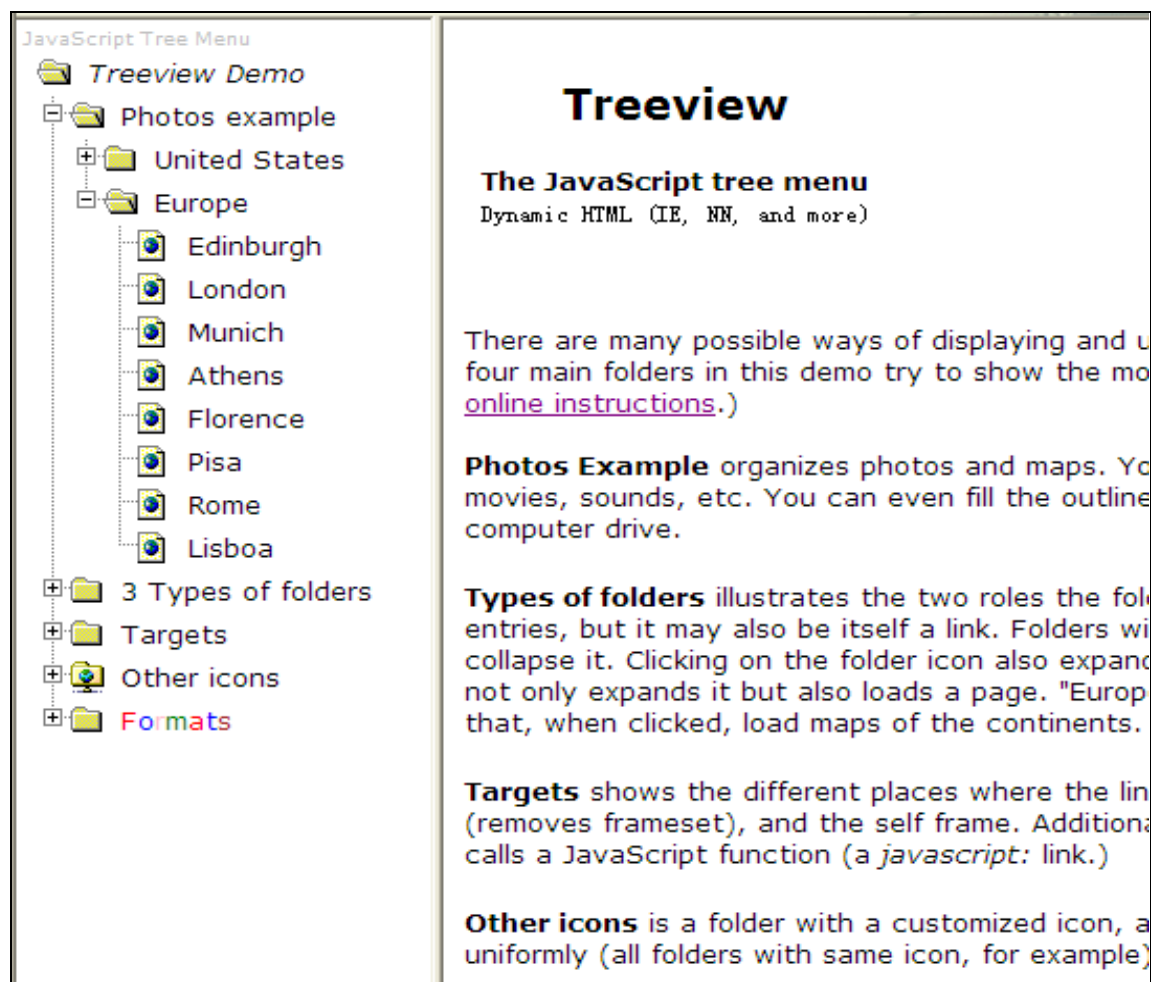
目 录

第 1 章 什么是TREEVIEW?	4
第 2 章 使用 TREEVIEW	8
第 3 章 安装 TREEVIEW	9
3.1 网页结构	9
3.2 可配置变量	10
3.3 使用Favorites Manager	13
3.4 手工创建树形菜单	14
3.4.1 gFld 函数	14
3.4.2 gLnk 函数	15
3.4.3 insFld 函数	16
3.4.4 insDoc 函数	17
3.4.5 综合使用这些函数	17
3.4.6 引号字符	18
3.5 可供选择的 API	18
第 4 章 高级配置	20
4.1 改变图标	20
4.2 配置风格	21
4.3 为节点指定额外的HTML	21
4.4 TreeView API	22
4.5 用菜单的ID来工作	22
4.6 用节点的ID来工作	23
第 5 章 从数据库来创建	25

第1章 什么是 TreeView?

什么是 TreeView?让我们先来看看它长什么样子。

下面是基于框架使用 TreeView 的图片。



下面是无框架使用 TreeView 的图片。

JavaScript Tree Menu

Tree Options

- [-] Expand for example with pics and flags
 - [-] United States
 - Boston
 - Tiny pic of New York City
 - Washington
 - [-] Europe
- [-] Types of node
 - [-] Expandable with link
 - [-] Expandable without link
 - Opens in new window

Frameless Layout For Treeview

This is the demo for the frameless layout and others check...

For simplification of the demo, most nodes are links (only with different arguments for different pages, and many users can jump from page to page clicking on the links).

Note how some of the expandable/collapsible nodes (the folder itself can load a page) are used for demo purposes.

下面是无框架使用 TreeView（高亮显示节点）的图片。

JavaScript Tree Menu

Tree Options

- [-] United States
 - Boston
 - Tiny pic of New York City
 - **Washington**
- [-] Europe
 - London
 - Lisbon



Testing highlight of selected node (a)

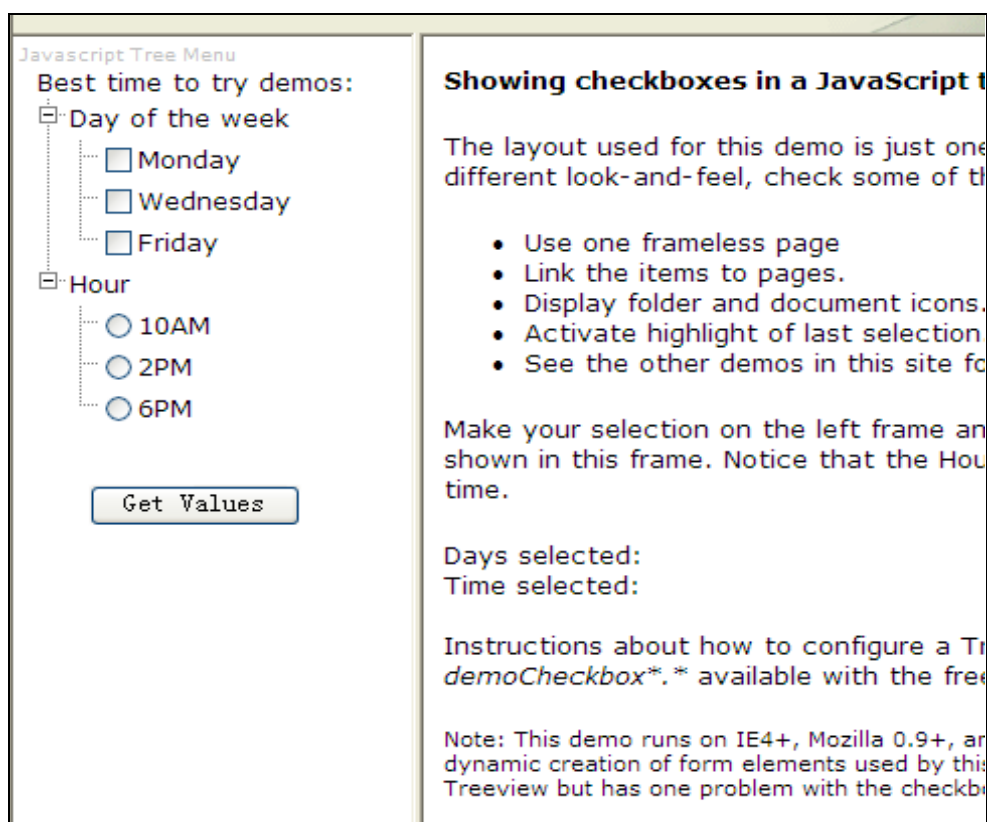
This demo is based on the [Frameless Layout](#) information on frameless layout configuration.

Two options are being tested here: the opening of the tree with all branches expanded or collapsed can be set on or off independently of each other for different based layouts. They are only being set for this demonstration.

Note how clicking on nodes that are the selected node (the United States) highlights the text of the node. Clicking on other nodes does not load a page (such as Europe).

Note also that, in the case of this demo, the selected node does not load a page. Webmasters prefer it that way. Not recommended for production use.

下面是基于框架使用 TreeView(在节点里使用表单元素)的图片。



现在应该知道 TreeView 是什么了吧？对，它就是一个用 Javascript 开发的树形菜单。

TreeView分为两种版本，一种是收费，一种是免费。这两个版本的功能完全一样，唯一不同的是，免费版本的菜单上方必须要有一个连接到www.treeview.net的网址。为了去掉这个很不爽的网址，我们只需要修改一下包含有TreeView的页面就可以了。具体如下：

```
<!----->

<!-- IMPORTANT NOTICE:                                -->

<!-- Removing the following link will prevent this script -->

<!-- from working. Unless you purchase the registered -->
```

```
<!-- version of TreeView, you must include this link.      -->

<!-- If you make any unauthorized changes to the following  -->

<!-- code, you will violate the user agreement.  If you want -->

<!-- to remove the link, see the online FAQ for instructions -->

<!-- on how to obtain a version without the link.          -->

<!------->

<TABLE border=0 style="display:none"><TR><TD><FONT size=-2><A
style="font-size:7pt;text-decoration:none;color:silver"
href="http://www.treemenu.ne/" target=_blank>Javascript Tree Menu 演示
</A></FONT></TD></TR></TABLE>
```

其中给红色部分我们要加上的，加上以后，这个表格就不显示出来了。而对于其它功能则没有任何影响。

第2章 使用 TreeView

TreeView 是一个简单并且快速的站点导航菜单。这一节将提供一些在页面上使用 TreeView 的技巧。

一般在使用 Treeview 的 WEB 网页上会包括两个或多个分隔区域。这些区域要么是表格，要么是框架。左边的区域一般显示树形的文件夹和文档节点，这就是树形导航菜单。这种层次结构可以装下很多的连接,但每次却只需要显示一点点的连接。单击减号 (-) 或加号 (+) 就可以马上展开或收缩这个文件夹节点。单击一个文档节点就可以在右边的区域加载一个新的页面或者打开一个新的窗口。

这里有一些在网页上使用 TreeView 的技巧。

- 在包含 frames 的站点里，拖动这个可见的分隔条就可以控制菜单的可用空间。
- 要访问隐藏的连接，单击文件夹或者前面的加号 (+)。这将会打开一个文件夹，显示它的子文件夹和里面的文档节点。
- 文件夹节点本身也可以包括一个连接。像这种文件夹节点，单击不仅可以展开的包括的内容，也可以同时打开它的指向的连接。
- 要收缩一个文件夹节点，单击 (-) 号。你将会看到其它文件夹里的其它的连接，而不用上下滚动。
- 单击最顶层的文件夹节点将会返回到入口页面。

第3章 安装 TreeView

1. 下载 [TreeView](#) 并解压ZIP文件到你的机器上。
2. 用你的浏览器打开 `index.html` 并且试用这些demos.
3. 选择最适合你的需求的demo, 编辑这个demo的文件。
请看这个 `readme.html` 文件 它对你下载的每个文件都有详细描绘。
4. 当你准备测试你编辑后的文件时, 你要复制或上传这些文件到你的 web 服务器上并且测试它。

注意: 如果你想在你的应用里使用demo, 并且想重命名这些文件, 你就需要搞清楚这些文件间的依赖关系。例如, 你想重命名 `demoFramesetRightFrame.html`, 你就同时还要更新 `demoFramesetNodes.js` 和 `demoFrameset.html`! 这个 `demoFrameset.html` file (这个文件对访问者来说具有最好的可见性。) 可以被重命名而没有任务依赖问题。

3.1 网页结构

在你的页面上使用 TreeView 有很多种选择。两种最常见的是:

- 将 TreeView 放在一个 frame 里, 并且这个 frame 只创建和显示这个菜单。
- 将 TreeView 放在一个无 frame 的页面里。

在这两种用法间有着一些细微的不同。 这些不同包括你怎样在页面上包含 TreeView, 你怎样设置 TreeView 的配置变量。然而, 不管你用哪种方式, TreeView 都使用同样的结构。现在我们来描绘一下这个公用的结构。

在包含菜单的每个页面上, 它的head块都包含以下内容:

- 一个指定样式的style标记。你可以用样式来改变菜单的外观。更多信息, 请参考下一节。

- 一个用来检测浏览器脚本的 `script` 标记 (`ua.js`).
- 一个载入 Treeview 引擎脚本的 `script` 标记 (`ftiens4.js`).
- 一个载入你的菜单配置文件的 `script` 标记 (例如 `demo*Nodes.js`.)

当然, 其它一些和菜单无关的语句也可以在 `head` 里。

在包含菜单的每个页面上, 它的 `body` 块应包含实际渲染菜单的语句。这是通过调用 `initializeDocument` 函数实现的。对于在菜单旁边还可能有其它元素的无框架页面来说, 这个 `initializeDocument` 执行时, 具体的相对位置则依赖于页面的布局。

注意, 如果从服务器端创建菜单代码, 那么结构会有一些不同。菜单的配置文件是放在 HTML 页面的开始部分, 而不是从一个单独的 JS 配置文件加载。

3.2 可配置变量

TreeView 包含了一些可配置的变量, 这些变量能够非常容易地让你使用或停用某些功能。这些 Treeview 的可配置变量应出现在 TreeView 的配置文件里 (例如可能叫 `demo*Nodes.js`)。这些配置变量应在节点配置代码的上面。如果你没有设置任何配置变量, 那么 TreeView 将使用这些变量的初始值。可配置变量的说明见下表。

变量名	描述	初始值
BUILDALL	对于很大的菜单来说, 高性能是通过延迟创建节点来获得的, 只有当用户展开这个文件夹时才创建它里面所包含的项目。默认情况下, TreeView 首先只创建必须的节点, 当某个文件夹被打开时才创建这个文件夹的其它节点。将这个变量设为 1 可以配置 TreeView 为在页面被加载时创建所有的节点, 即使那些节点是不可见的。你可以给这个变量赋予下列值中的任何一个:	0
	0: 延迟创建隐藏节点	
	1: 当页面载入时创建所有节点	

	通常, 只在一个 demo 里才设置这个变量; 这是一个显示复选框和单选按钮的例子, 它使用表单元素来作为节点, 因此, 当页面载入时, 整个菜单必须被创建, 以确保整个表单被创建.	
GLOBALTARGET	如果一个菜单被创建时使用了 addChildren API, 那么这个变量就可以指定菜单中所有连接的 target 属性. 你可以给这个变量赋予下列值中的任何一个: 'B': 设置 target 为新开窗口 'R': 设置 target 为右边的框架 'S': 设置 target 为当前框架 (替换这个菜单所在的页面) 'T': 设置 target 为当前浏览器窗口 (替换整个框架, 如果存在的话)	'R'
HIGHLIGHT	当节点被选中时, 这个变量将启用或停用节点高亮显示功能. 高亮显示将会反转当前被选中节点的背景和前景色. 除了你自定义这个行为 (请参考 the HIGHLIGHT_COLOR 和 HIGHLIGHT_BG 变量) 以外. 注意, 高亮只工作在那些载入页面的节点上. 你可以给这个变量赋予下列值中的任何一个: 0: 不高亮显示选取的节点 1: 高亮显示选取的节点	0
HIGHLIGHT_BG	这个变量指定了当高亮显示被启用时的背景色. 要查看更多关于高亮的信息请看 HIGHLIGHT 变量. 你可以给这个变量赋予任何有效的 HTML 颜色值, 如 'blue' 或 '#ccff99'.	'blue'
HIGHLIGHT_COLOR	这个变量指定了当高亮显示被启用时的文本色. 要查看更多关于高亮的信息请看 HIGHLIGHT 变量. 你可以给这个变量赋予任何有效的 HTML 颜色值, 如 'white' 或 '#FFFF80'.	'white'
ICONPATH	如果你想把 TreeView 的图片放到一个单独的目录下, 你就可以使用这个变量来设置那些图片的位置. 这个变量的值, 要是么一个空字符串 ("), 要么是一个有效的指向一个目录的 URL, 并且应以 '/' 终止. 一个空的字符串表明图片和 TreeView 文件是在同一目录下. 一些可能的值是: 'images/', 'http://www.x.com/y/', 等等.	"

<p>PRESERVESTATE</p>	<p>用户可以通过两种方式来改变菜单的状态：通过打开和收缩文件夹，能过高亮显示节点（如果 HIGHLIGHT=1）。除非状态信息被存贮，否则当重新载入这个菜单时，它将恢复到它的初始状态。通过将这个变量设置为 1, Treeview 可以存贮状态到 cookies 里，当页面再次载入时可以保持菜单的状态。你可以将下面的值赋予给这个变量：</p> <p>0: 当页面载入时不保存菜单的状态</p> <p>1: 存贮状态到 cookies 里，下次访问时使用这个状态</p> <p>不要将 PRESERVESTATE=1 和 STARTALLOPEN=1 在一起使用。这些设置是相互冲突的。</p> <p>有两种典型的特定情节对于保持菜单状态是不适合的，无框架布局（查看 USEFRAMES），和数据库驱动的应用。如果你的菜单的结构在重载时可能会改变（比如，在一个数据库驱动的应用里有一个新的文件夹被增加）而你又想使用 PRESERVESTATE，你的节点就必须要有个不变的标识。否则，脚本可能会使用一个错误的 ID 并且显示一个错误。要得到更多信息，请看下面的设置扩展 ID 这一节。</p> <p>如果你的站点使用了不止一个 TreeView，你就需要一种方法可以唯一标识不同的 TreeView，以作为 PRESERVESTATE 的机制。请看下面的设置 ID 这一节。</p>	<p>0</p>
<p>STARTALLOPEN</p>	<p>这个变量可以设置在载入页面之前，是否菜单里所有的文件夹都打开了。你可以将下面的值赋予给这个变量：</p> <p>0: 只展开根节点，不展开其它文件夹。</p> <p>1: 展开所有文件夹，显示菜单里的每一个节点。</p> <p>不要将 PRESERVESTATE=1 和 STARTALLOPEN=1 在一起使用。这些设置是相互冲突的。</p> <p>注意，展开所有节点可能只适合于小型或中型菜单，对于大菜单（比如有成百上千的节点。）来说，它将使前面介绍的优化毫无用处。</p>	<p>0</p>

<p>USEFRAMES</p>	<p>这个变量指定菜单是出现在一个框架里, 还是出现在一个正常的无框架页面里。如果你使用一个无框架布局, 一定要确保明确地设置了这个变量。否则 TreeView 将不会如你所愿地运行。你可以将下面的值赋予给这个变量:</p> <p>0: 在一个无框架布局里使用 TreeView</p> <p>1: 在一个框架布局里使用 TreeView, 并且菜单在它自己的框架里。</p>	<p>1</p>
<p>USEICONS</p>	<p>这个变量设置 TreeView 是否在菜单里显示图标。将它设为 0 则会删除这些图标。注意, 不管怎么样, + 和 - 总是会出现, 只是文件夹的外观和节点图标被配置。你可以将下面的值赋予给这个变量:</p> <p>0: 不显示图标</p> <p>1: 显示图标</p>	<p>1</p>
<p>USETEXTLINKS</p>	<p>一个节点的图标也可以连接到这个节点的目标。这个变量配置节点的文本是否也能连接到节点的目标。你可以将下面的值赋予给这个变量:</p> <p>0: 图标是唯一的连接</p> <p>1: 图标和文本都作为连接</p>	<p>0</p>
<p>WRAPTEXT</p>	<p>这个变量设置每个节点里的文本是否允许转行显示。你可以将下面的值赋予给这个变量:</p> <p>0: 不转行, 无论多长都只显示为一行</p> <p>1: 转行显示</p> <p>注意: 如果你是以使用框架的那个 demo 为基础来设计你的应用的, 请不要忘记删除左边框架 HTML 文件顶部的 style 标记中 nowrap 设置。</p>	<p>0</p>

3.3 使用 Favorites Manager

要节省创建菜单的时间, 你可以使用 Favorites Manager 程序 (by GubuSoft). Favorites Manager 本质上是一个连接管理工具. Favorites Manager 允许你通过对话框来输入连接的名称和 URL. 你可以用 Favorites Manager 对话框来增加, 修改, 删除, 和导出文件夹和连接. 当你准备将连接输出到 TreeView 时, 你可以以

TreeView能够使用的一种特殊格式从Favorites Manager导出这些连接. 关于使用Favorites Manager来可视化地创建菜单的详细介绍请看 [TreeView开发者资源](#).

3.4 手工创建树形菜单

3.4.1 gFld 函数

这个 gFld 函数允许你为菜单创建一个文件夹.调用这个函数需要两个参数并且返回一个文件夹:

参数	描述
Title	指定出现在文件夹名里文本.这个文本可以包含简单的 HTML 标记,例如格式标记(i, b, div, 等等). 如果你想在节点名上放一个小图标的话,它甚至可以包含一个 img 标记, 例如一个"new!" 图标.
Link	指定一个可选的 URL.这个 URL 可以是一个简单的文件名如 demoFramesetRightFrame.html 或者是一个协议字符串,域名,路径,和文件名比如 http://chinesedocument.com/index.html.

你可能不想单击一个文件夹节点来打开一个页面. 既然如此,你可能也不想单击文件夹前的 '+' 号来打开一个页面. 这就是说,你可能只想单击文件夹名来展开这个文件夹. 要做到这点,你需要给 gFld 函数的第二个参数赋予这样的值 javascript:parent.op() . 第二个参数调用一个空的函数,这样做是为了支持老版本的浏览器. 请看下面的例子:

```
aux1 = insFld(foldersTree, gFld("Test A", "javascript:parent.op()"))
```

如果使用了 javascript:parent.op() 这个参数,但文件夹没有正确地运行,请检查确保这个函数被定义在包括TreeView的框架HTML文件的header里.

3.4.2 gLnk 函数

这个 gLnk 函数允许你创建一个连接.这个函数仅仅被用来创建到文档的连接.至于这些连接是怎么出现在菜单里,是由其它函数决定的.调用这个函数需要三个参数并且返回一个连接:

参数	描述
Target	配置这个连接的目标位置.可以指定下列值中的一个:
	"R": 在右边的框架里打开文档 (一个名字 basefrm 的框架)
	"B": 在新窗口打开文档
	"T": 在当前浏览器窗口打开文档,如果存在的话则替代这个框架集
	"S": 在当前框架打开文档,替代这个菜单
	注意: 这个参数是大小写敏感的;确保使用大写字符.
Title	指定出现在连接里的文本.这个文本可以包含简单的 HTML 标记,例如格式标记 (i, b, div,等等). 如果你想在节点名上放一个小图标的话,它甚至可以包含一个 img 标记, 例如一个"new!" 图标.
Link	指定文档的 URL. 可以是绝对的或相关的 URL. 不要在这个字符输入任何其它信息; 增加一个 target 参数或者是事件处理都将会导致脚本无法运行.

例如,下面的函数调用将会在菜单里创建一个文本为My File的连接,单击这个节点会在新的窗口显示<http://www.mysite.com/MyFile.doc>文件:

```
gLnk("B", "My File", "http://www.mysite.com/MyFile.doc")
```

你还可以扩展这个Target参数,来指定连接协议.这是通过在大写字符后面紧跟一个小写的字符来指定目标位置的.:

"h": 代表 http:// 协议

"s": 代表 https:// 协议

"f": 代表 ftp:// 协议

"m": 代表 mailto:// 协议

注意: 这个参数是大小写敏感的;确保是小写字符.

这个参数允许你省略连接参数里的协议字符,这将会保证文件尺寸尽可能地小. 使用这个扩展,前面的例子可以重写为 :

```
gLnk("Bh", "My File", "www.mysite.com/MyFile.doc")
```

3.4.3 insFld 函数

这个 insFld 函数将一个文件夹放到另一个文件夹节点的里面.这个 insFld 函数需要两个文件夹节点参数并且把其中一个文件夹放到另一个文件夹里. 这就是说,这个函数把一个子文件夹节点放到一个父文件夹节点里. 这个函数需要两个参数:

参数	描述
Parent Folder	指定一个父文件夹. 也就是说,这个参数指定你想要把另一个文件夹放到哪里.
Child Folder	指定一个子文件夹. 也就是说, 这个参数指定你想要把哪个文件夹放到另一个文件夹里.

例如,下面的代码将把gFld("Europe", "http...") 子文件夹节点放到foldersTree 父文件夹节点里:

```
aux1 = insFld(foldersTree, gFld("Europe", "http..."))
```

3.4.4 insDoc 函数

这个 insDoc 函数把一个文档节点放到一个文件夹节点里. 这个 insDoc 函数需要两个参数,并且把这个文档节点参数放到一个文件夹节点参数里. 这个函数需要下面两个参数:

参数	描述
Parent Folder	指定一个父文件夹. 也就是说,这个参数指定你想要把一个文档节点放到哪个文件夹节点里.
Document Node	指定一个文档. 也就是说, 这个参数指定你想要把哪个文档节点放到一个文件夹节点里.

例如,下面的代码将把gLnk("S", "Boston", "...") 文档节点放到 aux2父文件夹节点里:

```
insDoc(aux2, gLnk("S", "Boston", "..."))
```

3.4.5 综合使用这些函数

下面的步骤将告诉你如何综合使用这些函数来创建一个 TreeView:

1. 创建一个根文件夹. 例如:

```
foldersTree = gFld("<i>Demo</i>", "demoFramesetRightFrame.html")
```

2. 通过insFld 函数把其它文件夹放到这个根文件里. 例如:

```
aux1 = insFld(foldersTree, gFld("Europe", "http..."))
```

3. 通过 insDoc 函数把文档节点放到文件夹里. 例如:

```
insDoc(aux1, gLnk("S", "Boston", "..."))
```

3.4.6 引号字符

使用单引号 (') 和双引号 (") 需要非常小心. 当你在下面特定情节使用引号时可能会非常郁闷:

- 当节点文本包含了这些符号(引号, 译者注)时
- 当超连接有如javascript:function... 格式并且这个函数的字符参数包含了引号字符.

你可以使用 `String.fromCharCode()` 函数来确保正确使用引号. 同样, 你可以通过在引号前输入 `(\)` 来"escape" 引号 . 下面的代码显示了这两种方法:

```
foldersTree = gFld("jack's", 'javascript:alert("jack" + String.fromCharCode(39) + "s")')
insDoc(foldersTree, gLnk("R", "jack's", 'javascript:alert(\\'jack\\' + String.fromCharCode(39) + \'s\'))')
insFld(foldersTree, gFld("db quote:''", 'javascript:alert("db quote:" + String.fromCharCode(34))'))
insDoc(foldersTree, gLnk("R", "db quote:''", 'javascript:alert(\\'db quote:\\' + String.fromCharCode(34))'))
```

3.5 可供选择的 API

如果你的菜单有多于一千个节点, 那么你可能就要用服务端代码来生成TreeView. 如果是这样的话, 通过服务器生成的JavaScript代码就应该使用一个与上节描述不同的API. 这个可供选择的API被用于文档连接的创建, 在页面载入后还可以节省菜单被渲染的时间. 自从这个API被编写到程序里以后, 它仅仅对于非常巨大的菜单有显而易见的好处, `insDoc` 和 `gLnk` API 被推荐用于手工配置的小型菜单.

`folder.addChildren` 函数允许你使用一个单一的语句来声明和构建一个文件夹里所有的文档节点。如果这个文件夹还有子文件夹，那么这些子文件夹必须被首先创建，然后才调用 `addChildren` 函数。由于这个顺序依赖，使用 `addChildren` API 的菜单配置文件与那些使用 `insFld` 和 `insDoc` API 的有很大的不同。事实上，它们创建文件夹的顺序是相反的。

`addChildren` 是 `folder` 对象的一个方法。`folder` 对象必须在 `addChildren` 方法被调用之前被创建。`addChildren` 方法唯一的参数是一个 `children` 的列表。在这个列表里的每一个元素都是一个文件夹对象或一个子列表：一个用于文档节点的标签，而另一个则用于节点的连接。下面看看 `addChildren` API 的例子：

```
fldChild = gFld("FChild", "linkfld1.html")
fldParent = gFld("F", "linkfld2.html")
fldParent.addChildren([fldChild, ["Doc", "link.html"]])
```

通过这个代码,用户可以看到文件夹 `F` 有两个节点: 子文件夹节点 `FChild` 和文档节点 `Doc`。

要指定使用 `addChildren` 创建的连接的 `target`, 你需要使用 `GLOBALTARGET` 配置变量。更多关于 `GLOBALTARGET` 的信息请看上面的 [TreeView 可配置变量](#)

`addChildren` API 隐藏了文档对象的创建（也就是没有显示的创建文档对象，译者注）。但是，你还是可以使用那些对象(文档对象，译者注)来改变像 `xID` 和 `iconSrc` 等的属性。为了得到对文档对象的访问，可以使用文件夹对象的 `children` 数组成员。例如：

```
fldParent.children[0].xID = "foo"
```

这个免费下载的 `TreeView` 包含了一个使用这个 API 在服务端动态生成的 ASP 演示。那些文件都是叫 `demoDynamic*.*`。

第4章 高级配置

4.1 改变图标

TreeView 提供了一套默认的图标. 然而,你也可以使用你自己的图标来创建文件夹和节点. 当为图标创建图像时,你可以使用任何图像文件,只要宽是 16 像素,高是 22 像素.

如果你想替代文件图标,你需要两个图片:一个打开文件的图片,一个关闭文件的图片. 当然,如果你愿意,你也可以使用同一张图片(最好不要这样. 译者注). `iconSrc` 和 `iconSrcClosed` 属性用来覆盖节点的图标.

如果你看过下载包里的那些文件,你就会发现基于框架布局的那个TreeView demo显示了如何使用图标. 在文本编辑器里打开 `demoFramesetNodes.js` 文件. 下面是覆盖了文件夹图标的代码:

```
aux1 = insFld(foldersTree, gFld("Other icons", "javascript:parent.op()"))
aux1.iconSrc = ICONPATH + "diffFolder.gif"
aux1.iconSrcClosed = ICONPATH + "diffFolder.gif"
```

在这个代码里, `ICONPATH` 是一个TreeView的环境变量. 它表明了TreeView里图片的位置. 如果你没有在应用里设置这个环境变量(没有设置就表示为空,译者注),它就可以在上面的语句里省略掉. `diffFolder.gif` 文件是一个真实的文件夹图标(它随着TreeView一起分发,在TreeView.zip包里).

类似地, `demoFramesetNodes.js`里的代码覆盖了文档图标:

```
docAux = insDoc(aux1, gLnk("B", "D/L Treeview",
"http://www.treeview.net/treemenu/download.asp"))
docAux.iconSrc = ICONPATH + "diffDoc.gif"
```

对于文档节点图标来说,它是由 `iconSrc` 属性来指定的.

4.2 配置风格

菜单的风格是在 `style` 标记里配置的(在包含菜单的文件中的 `head` 块里)。

如果你在一个框架里配置菜单的外观风格 (只包含菜单的那个框架),你就可以用 `a` 和 `td` 等CSS来控制字体, 颜色等等。

如果你使用一个无框架的布局并且页面上还有其它一些非菜单的元素(例如, 其它的锚点元素 (`a`)), 你又只想为菜单指定风格, 你就应该看看那个无框架demo的文件里是怎么做的。在那些文件里, `style` 块分为了两个部分: 菜单风格和其它风格。为了渲染菜单连接的锚点标记, 我们增加了一个特殊的class名叫 `TreeviewSpanArea`的 `span` 块. 使用这个 `span` 块,我们只给菜单应用了风格。

4.3 为节点指定额外的 HTML

你使用 `gFld` 和 `gLnk` 函数可以给文件夹和文档节点指定文本。你还可以在文本里指定一些HTML元素, 这些都将传给那些函数。 你可以使用 `bold`, `italics`,等等标记. 你甚至可以指定包含图像的HTML.然而, 在 `...`块部分你可以指定的HTML是非常有限的。

如何在菜单里包含复选框和单选按钮? 这是一个问题。 为了解决这个问题, 我们增加了一个 `prependHTML` 成员. 这个成员允许你在图标和文本之间指定文本. `prependHTML` 可以用于文件夹和文档节点. 作为另一个对象成员, `prependHTML` 应该在对象被创建后进行正确的设置。 例如:

```
newObj = insDoc(aux1, gLnk("B", "Item 5", "..."))
newObj.prependHTML = "<td valign=middle>not part of link</td>"
```

或者:

```
newObj.prependHTML = "<td valign=middle><input type=checkbox id='CB1'></td>"
```

当使用

```
prependHTML
```

时需要遵守一个重要的规则。为了兼容一些老版本的浏览器，菜单是用

在使用 `STARTALLOPEN` 和 `BUILDALL` 时也有一个重要的规则。菜单里的所有表单元素在页面载入时都必须被创建，以便这个表单能包含他们。（想想，如果表单都没有创建完整，那肯定会出错的。）

如果你想看一个关于 `prependHTML`的例子，请看TreeView里那个显示复选框的 demo.

4.4 TreeView API

菜单里的节点，如文件夹或文档，是由JavaScript对象在内部产生的，它们甚至在菜单开始用HTML渲染之前就存在了。你不应该对这些对象和他们生成的DOM对象感到困惑。例如，一旦菜单被渲染，一个 `Treeview Folder` 对象可能会生成几个DOM对象 (`images`, `text`, 等等.)

如果你用节点ID给菜单里的节点指定了外部的ID，（参考 [用节点的ID来工作](#)），你就可以通过调用这个函数来得到一个节点对象

```
myNode = findObj(id)
```

一旦你有了一个Treeview节点对象的引用，也就是一个JavaScript变量,那么你就可以用它做很多事情。例如，你可以通过调用函数来高亮显示它 `highlightObjLink(myNode)`. 你还可以通过调用 `myNode.parentObj`来查找一个节点的父节点（也就是容纳这个节点的文件夹）。

4.5 用菜单的 ID 来工作

如果你的站点里有不止一个树形菜单，你就需要给每个树形菜单一个唯一的标识。这个标识符将帮助每个树形菜单独立地保持住它们的状态。使用 `treeID` 属

性就能给每个树形菜单赋予一个标识符。例如，下面的代码将赋予菜单一个名为 t3 的标识符：

```
foldersTree = gFld("Three years", "demoLargeRightFrame.html")
foldersTree.treeID = "t3"
```

4.6 用节点的 ID 来工作

默认情况下，节点是由它们在菜单里的顺序来标识的。连续的数字标识符被顺序地赋予给菜单里的节点。第一个文件夹是数字 1，这个文件夹里的第一个项目就是数字 2，依此类推。PRESERVESTATE 机制是通过把它们 ID 联合存贮在 cookies 里来实现的。

如果在一个会话期间菜单的结构有变化(也就是说，节点被增加或删除)，这些 ID 也就改变。这就意味着 cookies 里的那些节点 ID 不再有效，并且 PRESERVESTATE 机制也不再如你所愿地工作。为了保证不出现这种情形，你可以硬编码替代这些 ID。当你硬编码静态 ID 时，就不能再使用默认的动态 ID 了，你就能避免 cookies 里的那些 ID 成为不同步的 ID 了。这就是说，那些 ID 不再依赖于菜单的结构了。(如果菜单很多的话，怎么可能给每个节点去手工起一个名字呢？就算可以，我估计那些名字也是同数字脱不开关系的。那么，这同默认情况下自动生成的 ID 又有什么实质性的区别呢？译者注)

xID 属性允许你给节点赋予一个 ID。你可以给 xID 属性赋予一个字符或数字。如果你使用字符，那这个字符串就不能包含下列字符：

- ^
- '
- "

这里有一个根据 demoLargeNodes.js 文件的代码改写的使用 xID 的例子：

```
foldersTree = gFld("Root", "a.jpg")
foldersTree.xID = "B8A4"
aux1 = insFld(foldersTree, gFld("Folder 1", "b.jpg"))
aux1.xID = "1A2"
aux2 = insDoc(aux1, gLnk("R", "Doc 1", "c.jpg"))
aux2.xID = "XYZ"
```

一定要注意:

- 如果你使用 `xID`, 菜单里所有节点都必须使用这个属性.
- 如果你使用 `xID`, 你必须给根节点赋一个值:
`foldersTree.xID = ...`
- 要注意变量和属性名的大写。比如你使用 `xId`, 那它就不会运行。
- 每个节点 `xID` 值必须是唯一的。
- `xID` 的值必须是一个常数, 不变的。
- 菜单里一个节点的 `xID` 不应该依赖于其它节点。
- `PRESERVESTATE` 变量和 `xID` 属性对于保持文件夹节点的状态和节点的高亮显示状态是非常重要的。如果你不需要节点高亮显示, 你可以简单地只给文件夹赋予 `ID`。(从这里是不是可以理解为, `PRESERVESTATE` 仅仅保存文件夹节点的状态, 是开还是关, 而用户最后一次点击的是文件夹下哪个文档节点, 它并没有保存。要想保存这个, 就必须用上高亮显示功能。
译者注)

如果在一个用户访问期间, 菜单的结构发生了变化, 这就说明这个菜单可能是在服务器端驱动的应用, 而且使用了数据库。如果是这样, 有一个比较好的用法是, 使用数据库的主键来作为 `xID` 的值。

第5章 从数据库来创建

如果你想在你的服务器或网络里创建一个树形菜单来显示目录和文件,我们已经特别为你制作了一个产品: 购买我们的 [Web File Browser](#).

如果你想从数据库里读取内容来创建 Treeview, 那就请继续往下看吧。

在你尝试使用Treeview进行任何服务端编码之前, 请一定要读 [网页结构](#) 和 [高级配置](#) 这两节.通过服务端动态生成的TreeView代码同一个静态的TreeView有着相同的结构, 并且使用同样的函数和配置变理来构建。 如果你将要创建的TreeView有非常多的节点, 请一定要仔细阅读[可供选择的 API](#) 这一节。

在阅读并理解了上面这些节以后, 你就可以看看下载的那个动态的demo了。 那些文件都叫 demoDynamic*.*. 这个代码给你提供了一个很好的起点, 即使你的工程既不用ASP也不用Access数据库。就像你看到的那样, 这个ASP文件有两个角色: 它生成了一个包含树形菜单的正常HTML的页面, 同时也生成了JavaScript, 和独立的静态的 demo*Nodes.js 配置文件里的内容一样。

为了产生必要的 gFld 和 gLnk 语句, demoDynamicLeftFrame.asp 文件里的 outputJavascript 函数查询了数据库。 请注意每次调用 response.write 都是加了一个换行符的。这是一个公共误区, 就是在两行JavaScript语句间忘记换行符。