大纲模式

Emacs 的大纲模式(Outline mode)，是一个十分有用的模式。如果工程规模比较大，你应该用大纲来组织它。

大纲模式通常作为辅模式使用，**M-x outline-minor-mode**启用。

大纲模式可以根据代码的语法对结构进行识别，但是这种自动模式工作的不是很好，而且不够灵活

另一种工作方式是查找特定的字串，来组织文档的结构。这种工作方式是可控制的，不过需要手动加入这些作为结构标识的字串。似乎有点麻烦，但是对于严谨的构思来说，你是需要列一个提纲的

大纲模式默认以行首的 * 作为结构标识，每多一个 * 就是下一级分支。

大纲模式默认的键绑定太过复杂，在配置文件中添加以下语句，将所有大纲模式操作的前缀键改为 Ctrl+o

```
(setq outline-minor-mode-prefix [(control o)])
```

试着在文档中随便加入几个这样的标识

```
*主干一**分支一**分支二*主干二
```

进行一些简单的操作

C-o C-a	全部显示
C-o C-t	显示主干
C-o C-q	全部隐藏
C-o C-i	显示下一级分支
C-o C-k	显示分支
C-o C-e	显示节点
C-o C-d	隐藏分支

定制结构标识

为了不影响代码的功能，通常要把结构标识放到注释中，这样作有可能会带来不便,例如在 XML 环境中就要像这样使用：

```
<!--*结构标识-->
```

这样会给你的工作制造出一些混乱，并且大纲模式是以行为单位进行识别的，也就是说**<!--**属于上一个分支

好在大纲模式可以通过正则表达式来定义结构标识，你可以把结构标识定义为下面这种形式

```
<!--*结构标识-->
```

通过设置**outline-regexp**定制标识：

```
setq outline-regexp "<!\\❶-❷\\-\\*+❸
```

❶ Elisp 语法中，**** 有特殊意义，所以脱字符要用****表示
❷ 正则表达式中 **-** 有特殊意义，所以前面要加脱字符
❸ ***+** 表示一个或多个*

配置

完整的配置：

例 25.2. emacs 大纲模式

```
(setq outline-minor-mode-prefix [(control o)])❶(require 'docbook-xml-mode)❷(add-hook 'docbook-xml-
```

❶ 更改大纲模式前缀键
❶ 加载 docbook-xml-mode
❸ 添加 docbook-xml-mode 钩子，运行下面代码[47]
❹ 将`<!--*`识别为标识
❺ 启动大纲模式作为辅模式
❻ 隐藏所有内容，只显示主干

操作列表

显示、隐藏

	全部显示		显示分支		隐藏	
全局	show-all	C-o C-a	hide-body	C-o C-t	hide-sublevels	C-o C-q❶
分支	show-subtree	C-o C-s	hide-leaves	C-o C-l	hide-subtree	C-o C-d
			show-branches	C-o C-k		
			show-children	C-o C-i		
节点	show-entry	C-o C-e			hide-entry	C-o C-c
其它分支					hide-other	C-o C-o

❶ 可以带数字参数，如`M-2 C-o C-q`显示第2层子结构

移动

	向前		向后	
全局	outline-next-visible-heading	C-o C-n	outline-previous-visible-heading	C-o C-p
同级	outline-forward-same-level	C-o C-f	outline-backward-same-level	C-o C-b
返回上一级	outline-up-heading		C-o C-u	

[47] `docbook-xml-mode.el`中定义 docbook-xml-mode-hook，模式启动时运行钩子代码