以名称－值属性方式传递给tinyMCE的init方法的设置控制着程序的全局行为。所有的键及可能的键值如下表所示。注意，用[]括起来的名称是可选的，不是必须的设置。

常规设置

mode

Mode可以是以下几个值之一：

textareas - 页面加载时把所有TEXTAREA组件转换成编辑器.

specific\_textareas - 将所有"mce\_editable"属性值为true的TEXTAREA转换成编辑器.

exact - 只转换在"elements"设置中指定的确切组件.

[theme]

指定要使用的主题名称，主题将被放在TinyMCE的themes目录下，默认为default。TinyMCE自带三个内置的主题，它们是simple，default和advanced。

如果你想创建自己的主题，请仔细阅读文档的 主题 部分。

[plugins]

此选项是一系列以逗号分隔的主题插件列表（例如，可以只用来扩展图像对话框）。这些插件覆盖了主题模板中定义的功能。插件逻辑应该包含在一个叫"editor\_plugin.js"的文件中，它是插件目录中唯一被包含进来的文件。

例如："my\_image\_dialog,my\_link\_dialog".

如果你想创建自己的插件，请仔细阅读文档的 主题 部分。

[language]

TinyMCE中使用的语言包，这应该是像se,uk,us等诸如此类的FN代码，它被用来从"langs"目录中获取语言包，该设置的默认值为"uk".

中文用户建议您使用"zh-CN"

[elements]

以逗号分隔的用来转换成编辑器的组件列表，该选项仅在"mode"选项被设置成"exact"时使用。该列表中的元素可以是任何有id或name属性的HTML组件。

[ask]

当"mode"被设置成"textareas"或"specific\_textareas"的时候，该选项被应用，询问用户输入框是否将被转换成编辑器。

如果你想使用这个选项，将它设置为true.

[textarea\_trigger]

textarea（文本域）触发器的属性，默认值为"mce\_editable"。

该选项仅在"mode"被设置为"specific\_textareas"时使用。

[valid\_elements]

以逗号分隔开的组件转换部分的列表。

例如： a[href|target=\_blank],strong/b,div[align],br.

以上例子告诉TinyMCE移除所有除了"a, strong, div"和"br"的组件，将元素b转换成strong，默认target设为"\_blank"并保持href、target、align属性。在匹配组件及属性的名称时可以使用像\*,+,?这样的通配符。

字符：

,

各组件定义之间的分隔符。

/

两个同义组件之间的分隔符。第一个组件是会被用来输出的那个（即第二个组件被第一个替换）。

|

各属性定义之间的分隔符。

[

定义某组件的属性列表的开始符号。

]

定义某组件的属性列表的结束符号。

=

将属性的默认值设为特定值。例如："target=\_blank"

:

将属性的值强行设为特定值。例如："border:0"

<

校验某个属性的值。例如："target<\_blank?\_self"

?

属性校验值之间的分隔符，见上。

特殊变量：

{$uid} - 产生一个唯一ID号。 例如："p[id:{$uid}]".

此选项的默认值是以下模式：

"a[href|target],strong/b[class],em/i[class],strike[class],u[class],p[class|align],ol,ul,li,br,

img[class|src|border=0|alt|hspace|vspace|width|height|align],sub,sup,blockquote[dir|style],

table[border=0|cellspacing|cellpadding|width|height|class|align],tr[rowspan],

td[colspan|rowspan|width|height],div[class|align],span[class|align],pre[class|align],

address[class|align],h1[class|align],h2[class|align],h3[class|align],

h4[class|align],h5[class|align],h6[class|align],hr".

要包含所有组件及属性，请使用 \*[\*] ，这在使用invalid\_elements选项时特别有用。

[extended\_valid\_elements]

向"valid\_elements"列表末尾添加可用组件。此选项在你仅仅想往默认列表中增添部分组件时相当有用。

格式与"valid\_elements"一致。

[invalid\_elements]

输出时需要排除在外的组件名称列表，以逗号分隔开。

[trim\_span\_elements]

True/False选项。如果设置为true，不需要的组件将被移除。默认值即为true。

[verify\_css\_classes]

True/False选项。如果设置为true，将会校验CSS的class属性。默认值即为true。

[verify\_html]

True/False选项。代表着HTML内容是否需要校验。默认值为true。

[urlconvertor\_callback]

当清理进程处理URL的时候调用的函数名。此函数必须遵从以下的格式：func(url, node, on\_save) ，返回转换后的URL。此设置专为集成目的而设。参数url代表要转换的地址，node代表包含URL的那个节点，on\_save是一个布尔值（在用户提交表单时为真）。

[preformatted]

True/False选项。如果设置为true，编辑器将把制表符（TAB）转换成缩进，除此外保持其他whitespace（空白字符、换行等）字符，就如同HTML标签中的PRE的效果。默认值为false。

[insertlink\_callback]

当执行"insertlink"命令时调用的函数名。此函数获取选定链接的地址和目标，返回一个以"href", "target"和 "title"为collection名称的数组。当使用新窗口时，为了兼容Mozilla，会在window.opener（弹出窗口的母窗口，译者注）上调用tinyMCE.insertLink。

[insertimage\_callback]

当执行"insertimage"命令时调用的函数名。此函数获取选定图像的url并返回一个以src和alt为collection名称的数组。当使用新窗口时，为了兼容Mozilla，会在window.opener上调用tinyMCE.insertImage。

函数格式：insertimage(src, alt, border, hspace, vspace, width, height, align, title, onmouseover, onmouseout, action).

[setupcontent\_callback]

当编辑器初始化时调用的函数名。函数格式：setupContentCallback(editor\_id, node)，其中editor\_id是编辑器的id，node是编辑器所在的body组件节点。

[save\_callback]

当执行triggerSave（触发保存）命令被调用时调用的函数名。函数格式：save(id, content, node)，如果有特定返回值，其值将会被加到HTML表单组件上。所以可以通过此函数来自定义用户转换逻辑。

[docs\_language]

TinyMCE文档中使用的语言，这应该是像se,uk,us等诸如此类的FN代码，它被用来从"<theme>/docs/<lang>"目录中获取文档。 该选项的默认值跟language选项一致。

[width]

编辑器的宽度，它默认的宽度是原先被替换的组件宽度。

[height]

编辑器的高度，它默认的高度是原先被替换的组件高度。

[content\_css]

编辑窗口中要使用的CSS文件，其路径应该跟页面相关。

[popups\_css]

像插入链接和图片时的弹出窗口中使用的CSS文件，其路径应该跟页面相关。

[editor\_css]

编辑器使用的CSS文件，其路径应该跟页面相关。

[encoding]

编辑器的输出编码，此选项目前只能为"html"或者空。如果设置为"html"，编辑器的输出将经过HTML编码处理。

例如：<将会变成&lt;，依此类推。默认值是空。

[debug]

True/False选项。如果设置成true，像css文件路径等调试信息将会被显示。

[visual]

True/False选项。如果设置成true，当边框设置成0的时候，为达到更好的视觉效果在编辑器中表格会有虚线出现。

默认值为true。

[visual\_table\_style]

用户可自定义表格的样式，默认值为："border: 1px dashed #BBBBBB"。

[add\_form\_submit\_trigger]

True/False选项。如果设置成true，将强制进行所有表单的"onsubmit"事件处理并引发保存。此选项默认值为true。

[add\_unload\_trigger]

True/False选项。如果设置成true，在"onunload"事件发生时，当前窗口会引发一个triggerSave调用。此triggerSave调用不会做任何清理工作，因为它是用来处理 前进/后退 按钮的。此选项默认值为true。

[force\_br\_newlines]

True/False选项。此选项强制编辑器将段落符号(P)替换成换行符(BR)。此选项默认值为false。（实验阶段）

[force\_p\_newlines]

True/False选项。如果打开此选项，在按下回车键（Enter）时Mozilla/Firefox浏览器会生成段落符号(P)，在按下Shift+Enter时会生成换行符(BR)。此选项默认为true。

[relative\_urls]

True/False选项。如果设置成true，绝对路径将被转换成相对路径。默认值为true。

[remove\_script\_host]

True/False选项。在设置成true的情况下，如果URL中的主机名、端口号跟编辑器当前所在一致，它们会被移除。

例如：编辑器所在站点为 http://www.somesite.com ，那么以下链接 http://www.somesite.com/somedir/somepage.html 将会被转换成 /somedir/somepage.html 。

如果relative\_urls被设置成false，此选项默认为true。

[focus\_alert]

True/False选项。如果设置成true，编辑器在失去焦点的时候就会出现一个烦人的警告框。默认值为true。

[document\_base\_url]

在将绝对路径转换成相对路径的时候会用到此文档的URL。这个选项指定了编辑器当前的默认文档。

注意如果此选项指定到一个域名，请添加协议前缀，并以斜杠结尾。例如：http://www.somehost.com/mydir/

[custom\_undo\_redo]

True/False选项。此选项可以使 撤消/重做 功能更加完善。默认值为true。

[custom\_undo\_redo\_levels]

自定义最多可撤消操作的次数，默认是无限制。

[custom\_undo\_redo\_keyboard\_shortcuts]

如果使用，编辑器可以使用Ctrl+Z和Ctrl+Y快捷键来实现撤消和重做。默认允许。

[fix\_content\_duplication]

True/False选项。此选项修正了MSIE中一个内容重复的bug。默认启用，但是为了兼容性也可以禁用(false)。

[directionality]

此选项可以设置像阿拉伯语等语言的文字方向。可能的值为：ltr, rtl。默认值: ltr（从左到右）。

[auto\_cleanup\_word]

如果启用，从MS office/Word粘贴的HTML将会自动被清理。此选项默认值为false。

注意：此操作当前仅支持MSIE。

[cleanup\_on\_startup]

如果启用，在编辑器初始化时文本域及组件将会被清理。默认值为false。

[inline\_styles]

如果启用，像width,height,vspace,hspace和align等属性将会被风格属性替代。默认值为false。

在使用此选项时记得要把风格属性应用在正确的组件上。

[convert\_newlines\_to\_brs]

如果启用，所有的\n(新行)将会在编辑器启动时被转换成<br />组件。此选项默认值为false。

[auto\_reset\_designmode]

因为在调用style.display的none/block方法来显示/隐藏某个TinyMCE编辑器的时候有bug，设计模式需要被重置。此选项一旦启用，当编辑器聚焦时，将自动重置。默认值为false。

[entities]

一张以字符代码来查找名称的表，该表中元素是以逗号分隔开的实体列表。该列表以奇数项、偶数项区分，其中奇数项是被用来转换的字符代码，偶数项是代表那个字符代码的实体名称。例如："8205,zwj,8206,lrm,8207,rlm,173,shy"。

[cleanup\_callback]

自定义清理函数。此选项能让用户在默认的清理基础上有所扩展。此函数调用跟默认的调用是分开的，它并不替换默认的清理函数，而仅仅是扩展。点击 插件清理 查看更多细节。

主题的高级特定设置

[theme\_advanced\_toolbar\_location]

此选项用来改变工具栏的默认位置。可能的值为："top"和"bottom"。默认值是"bottom"。

[theme\_advanced\_toolbar\_align]

此选项用来设置工具栏的对齐方式是left, center还是right。默认值是center。

[theme\_advanced\_styles]

此选项可以用来增加风格下拉框中的CSS类和名称。格式如下："<title>=<class>;.."。

如果没有指定此选项，主内容部分的CSS类将会被自动导入。

例如："Header 1=header1;Header 2=header2;Header 3=header3"

[theme\_advanced\_buttons1]

工具栏上第一行要包含的按钮列表，以逗号分隔开。例如："bold,italic,underline"。

允许的按钮名称如下：

bold, italic, underline, strikethrough, justifyleft, justifycenter, justifyright, justifyfull, styleselect, bullist, numlist, outdent, indent, undo,redo, link, unlink, image, cleanup, help, code, table, row\_before, row\_after, delete\_row, separator, rowseparator, col\_before, col\_after, delete\_col, hr, removeformat, sub, sup, formatselect, fontselect, fontsizeselect, forecolor,charmap,visualaid,spacer,cut,copy,paste

[theme\_advanced\_buttons2]

同上，差别在于指定的是工具栏第二行。

[theme\_advanced\_buttons3]

同上，差别在于指定的是工具栏第三行。

[theme\_advanced\_buttons<N>\_add]

向工具栏上特定的第N行中增加额外的控制/按钮。例如：theme\_advaned\_buttons3\_add : "iespell".

[theme\_advanced\_buttons<N>\_add\_before]

向工具栏上特定的第N行的默认按钮前面增加额外的控制/按钮。例如：theme\_advaned\_buttons3\_add\_before : "iespell"。

[theme\_advanced\_disable]

要禁用的按钮/组件的列表，以逗号分隔。例如："formatselect".

[theme\_advanced\_source\_editor\_width]

源文件编辑器窗口的宽度。

[theme\_advanced\_source\_editor\_height]

源文件编辑器窗口的高度。

[theme\_advanced\_path\_location]

组件路径列表的位置，可能的值为："top"或"bottom"。默认值为："none"

[theme\_advanced\_blockformats]

formatselect列表要屏蔽的格式列表，以逗号分隔开。默认值：p,address,pre,h1,h2,h3,h4,h5,h6。

以下是一个较为复杂的初始化例子：

<!-- tinyMCE -->

<script language="javascript" type="text/javascript" src="jscripts/tiny\_mce/tiny\_mce.js"></script>

<script language="javascript" type="text/javascript">

tinyMCE.init({

mode : "exact",

theme : "mytheme",

language : "se",

elements : "elm1,elm2"

});

</script>

<!-- /tinyMCE -->

函数 （供高级使用）

TinyMCE有一个全局实例，它能提供一些可从页面调用的公用函数。

--------------------------------------------------------------------------------

语法： tinyMCE.triggerSave([skip\_cleanup]);

描述：进行清除操作，并将编辑器内容移回到表单域中。tinyMCE通过向表单提交方法中增加触发器来自动调用此函数。

参数：

[skip\_cleanup] - 禁用保存触发器的清除功能，默认为false。(可选)

返回： 无

--------------------------------------------------------------------------------

语法： tinyMCE.updateContent(form\_element\_name);

描述：将表单组件的内容转移到编辑器中，此操作的功能跟triggerSave()正好相反。当你想动态改变编辑器内容时可以使用此方法。

参数：

form\_element\_name - 要获取内容的组件所在表单的名称。

返回： 无

--------------------------------------------------------------------------------

语法： tinyMCE.execInstanceCommand(editor\_id, command, [user\_interface], [value]);

描述：此方法通过editor\_id参数找到某个编辑器实例，在此编辑器上执行一个命令。

参数：

editor\_id - 编辑器实例的ID或者被替换后的组件id/名称。

command - 要执行的命令。查看execCommand函数以获取更多细节。

[user\_interface] - 是否使用用户界面。

[value] - 执行命令时要传递的参数，例如：一个URL。

返回： 无

--------------------------------------------------------------------------------

语法： tinyMCE.execCommand(command, [user\_interface], [value]);

描述：此方法在选定编辑器中通过名称执行特定命令。

参数：

command - 要执行的命令，例如："Bold" or "Italic"。你可以通过此连接来查看Mozilla Midas spec。但是tinyMCE也有自身一些特殊的命令如下表所示：

mceLink

打开插入链接对话框并插入链接。

mceImage

打开插入图像对话框并插入图像。

mceCleanup

从HTML代码中移除不需要的组件和属性。

mceHelp

打开文档页面。

mceInsertTable

在鼠标位置插入表格，默认尺寸为：2×2。如果execCommand函数中有指定value参数，它的格式必须为name/value形式的数组，其中 name有以下选项：cols, rows, border, cellspacing, cellpadding。border（边框）的默认大小为：0。

mceTableInsertRowBefore

在当前鼠标所在位置之前插入一行。

mceTableInsertRowAfter

在当前鼠标所在位置之后插入一行。

mceTableDeleteRow

将当前鼠标所在行删除。

mceTableInsertColBefore

在当前鼠标所在位置之前插入一列。

mceTableInsertColAfter

在当前鼠标所在位置之后插入一列。

mceTableDeleteCol

将当前鼠标所在列删除。

mceAddControl

向编辑器添加组件控制，此编辑器的id/name由value指定。

mceRemoveControl

通过editor\_id名称来移除特定控制。value是要移除的editor\_id（编辑器ID），编辑器ID格式如下："mce\_editor\_<index>"。

也可以使用DOM组件的ID和表单名。

mceFocus

使以value的值为ID的编辑器获得焦点。编辑器ID格式如下："mce\_editor\_<index>"。也可以使用DOM组件的ID和表单名。

mceSetCSSClass

设置CSS类属性，或者在选择框中创建新的span。value的值是要给选定组件指派的css类名或者要新创建的span组件名称。

mceInsertContent

在当前鼠标所在位置插入value的内容。

mceReplaceContent

将当前选定部分替换为value中的HTML代码。{$selection}变量由当前选定部分的文字内容替换。

mceSetAttribute

为当前选定组件设置属性。此命令的value应该是name/value数组，其参数如下：

name - 要设置的属性名。

value - 要设置的属性值。

[targets] - 要增加属性的目标组件，默认为：p,img,span,div,td,h1,h2,h3,h4,h5,h6,pre,address。

使用示例：

tinyMCE.execCommand('mceSetAttribute',false,{name:'align',value:'right'});

mceToggleVisualAid

是否开启视觉帮助模式

mceAnchor

插入名称锚点。value是锚点的名称。

mceResetDesignMode

重置所有编辑器实例的设计模式状态。在Firefox中，当编辑器被放在制表符中或者用style.display="none/block"来控制隐显的时候，此命令非常有用。在编辑器重新显示时调用此命令。

mceSelectNode

选定value指定的节点/组件。同时此命令会使编辑器滚动到此组件所在位置。

mceSelectNodeDepth

从当前节点以深度关系选择指定的节点/组件。所以0值会选定当前聚焦的节点。同时此命令会使编辑器滚动到此组件所在位置。

user\_interface - 申明某命令是否会显示用户界面。True/False选项。

value - 要传递给命令的值。例如，插入链接时，它就是链接的URL。

返回： 无

示例：

<a href="javascript :tinymce.execCommand('Bold');">[Do bold]</a>

--------------------------------------------------------------------------------

语法： tinyMCE.insertLink(href, target);

描述：此方法在当前选定编辑器实例的鼠标位置插入一个链接，或者用新的数据来更新已有链接。当你创建自己的主题时，此方法应在插入链接的弹出窗口中的OK按钮按下时被调用。

参数：

href - 链接的地址/URL。

target - 链接的目标。

返回： 无

--------------------------------------------------------------------------------

语法： tinyMCE.insertImage(src, alt, border, hspace, vspace, width, height, align, title, onmouseover, onmouseout);

描述：此方法在当前选定编辑器实例的鼠标位置插入图像，或者用新的数据来更新已有图像。当你创建自己的主题时，此方法应在插入图像的弹出窗口中的OK按钮按下时被调用。

参数：

src - 图像地址。

alt - 图片的替代文字。

border - 图像边框。

hspace - 图像水平间距。

vspace - 图像垂直间距。

width - 图像宽。

height - 图像高。

align - 图像对齐方式。

title - 图像链接的标题。

onmouseover - 图像在鼠标放上去时的事件处理。

onmouseout - 图像在鼠标移开时的事件处理。

返回： 无

--------------------------------------------------------------------------------

语法： tinyMCE.triggerNodeChange();

描述：当编辑器发生外部变化时此方法被调用。然后此方法回调主题的"handleNodeChangeCallback"方法。

参数： None

返回： 无

--------------------------------------------------------------------------------

语法： tinyMCE.getContent();

描述：此方法返回当前选定编辑器中的HTML内容。如果没有任何选定编辑器，该方法返回null。

参数： None

返回：当前选定编辑器中的HTML内容或者null。

--------------------------------------------------------------------------------

语法： tinyMCE.setContent(html);

描述：此方法设定当前选定编辑器的HTML内容。

参数：

html - 要设定的HTML源代码。

返回： None.

--------------------------------------------------------------------------------

语法： tinyMCE.getEditorInstance(editor\_id);

描述：此方法通过editor\_id返回某个编辑器实例。

参数：

editor\_id - 要获取的编辑器实例。

返回：编辑器实例(TinyMCEControl)。

--------------------------------------------------------------------------------

语法： tinyMCE.importThemeLanguagePack([theme]);

描述：此方法导入主题中特定的语言包。自定义主题中可以调用此方法。

参数：

[theme] - 当前主题名称。此选项可选，默认为全局的"theme"设定。此参数在主题插件中非常有用。

返回： None.

--------------------------------------------------------------------------------

语法： tinyMCE.importPluginLanguagePack(plugin, valid\_languages);

描述：此方法导入插件中特定的语言包。自定义插件中可以调用此方法。

参数：

plugin - 当前插件的名称。

valid\_languages - 以逗号分隔的所支持语言包列表。

返回： None.

--------------------------------------------------------------------------------

语法： tinyMCE.applyTemplate(html);

描述：此方法替换主题中的设置及语言变量。

参数：

html - 要替换主题变量的HTML代码。

返回： 转换后的HTML代码。

--------------------------------------------------------------------------------

语法： tinyMCE.openWindow(template, [args]);

描述：此方法通过从指定的template参数中获取width, height及html数据来打开新窗口。数组参数args中包含要替换的变量名。

还有一些自定义的窗口参数：

mce\_replacevariables - 启用/禁用HTML文档中的语言/变量替换。默认启用。

mce\_windowresize - 启用/禁用弹出窗口的自动resize特征，默认启用。

参数：

template - Name/Value数组，以width, height, html和file为键。

[args] - 包含要替换变量名的Name/Value数组。模板文件中的键值变量被参数列表所替换。

例如：image.php?src={$src}被替换为：image.php?src=image.gif。

返回： None.

--------------------------------------------------------------------------------

语法： tinyMCE.getWindowArg(name, [default\_value]);

描述：此方法根据name返回窗口参数，它可以用在主题中的弹出窗口中，以获取对话框中的参数。

参数：

name - 要获取的窗口参数的名称。

[default\_value] - 窗口参数丢失时返回的默认值。

返回： 窗口参数的值。

--------------------------------------------------------------------------------

语法： tinyMCE.setWindowArg(name, value);

描述：此方法根据name设定窗口参数，例如它在某插件调用其他插件时非常有用。

参数：

name - 要设置的窗口参数。

value - 窗口参数中要设置的值。

返回： 窗口参数值。

--------------------------------------------------------------------------------

语法： tinyMCE.getParam(name, [default\_value], [strip\_whitespace]);

描述：此方法返回TinyMCE的配置参数。

参数：

name - 要获取的窗口参数名。

[default\_value] - 窗口参数丢失时返回的默认值。

[strip\_whitespace] - 如果为真，所有返回值中的空白字符将会被除去。默认：false。