

## KindEditor 使用方法

### 编辑器调用方法

1. 下载 KindEditor 最新版本。
2. 解压 zip 文件，将所有文件上传到您的网站程序目录下。例如：<http://您的域名/editor/>
3. 要显示编辑器的位置添加 TEXTAREA 输入框。注：id 必须指定唯一值，还有有些浏览器上不设宽度和高度可能显示有问题，所以最好设一下宽度和高度。宽度和高度可用 inline 样式设置，也可用编辑器属性传进去。

```
<textarea id="your_editor_id" name="content" cols="100" rows="8" style="width:700px;height:300px;"></textarea>
```

4. 该页面添加以下代码。

```
<script charset="utf-8" src="/editor/kindeditor.js"></script>
<script>
    KE.show({
        id : 'your_editor_id'
    });
</script>
```

除 id 之外还可以设置其它的属性，具体属性请参考[编辑器属性](#)。

### 编辑器属性

调用 **KE.show** 或 **KE.init** 时可以设置以下参数。

1. **id**  
TEXTAREA 输入框的 ID，必须设置。  
数据类型：String
2. **items**  
配置编辑器的工具栏，其中 "-" 表示换行，"|" 表示分隔符。  
数据类型：Array  
默认值：

```
['source', '|', 'fullscreen', 'undo', 'redo', 'print', 'cut', 'copy', 'paste', 'plainpaste', 'wordpaste', '|', 'justifyleft', 'justifycenter', 'justifyright', 'justifyfull', 'insertorderedlist', 'insertunorderedlist', 'indent', 'outdent', 'subscript', 'superscript', '|', 'selectall', '-', 'title', 'fontname', 'fontsize', '|', 'textcolor', 'bgcolor', 'bold', 'italic', 'underline', 'strikethrough', 'removeformat', '|', 'image', 'flash', 'media', 'table', 'hr', 'emoticons', 'link', 'unlink', '|', 'about']
```

### 3. **width**

编辑器的宽度，可以设置 **px** 或 **%**，比 **TEXTAREA** 输入框样式表宽度优先度高。

数据类型: **String**

默认值: **TEXTAREA** 输入框的宽度

注: **3.2** 版本开始支持。

### 4. **height**

编辑器的高度，只能设置 **px**，比 **TEXTAREA** 输入框样式表高度优先度高。

数据类型: **String**

默认值: **TEXTAREA** 输入框的高度

注: **3.2** 版本开始支持。

### 5. **minWidth**

数据类型: **Int**

指定编辑器最小宽度，单位为 **px**。

默认值: **200**

### 6. **minHeight**

数据类型: **Int**

指定编辑器最小高度，单位为 **px**。

默认值: **100**

### 7. **filterMode**

数据类型: **Boolean**

**true** 时过滤 **HTML** 代码，**false** 时允许输入任何代码。

默认值: **false**

注: **3.4** 以前版本的 **filterMode** 默认值为 **true**。

### 8. **htmlTags**

指定要保留的 **HTML** 标记和属性。哈希数组的 **key** 为 **HTML** 标签名，**value** 为 **HTML** 属性数组，**"**开始的属性表示 **style** 属性。

数据类型: **Object**

默认值:

```
{
  font : ['color', 'size', 'face', '.background-color'],
  span : ['style'],
  div : ['class', 'align', 'style'],
  table: ['class', 'border', 'cellspacing', 'cellpadding', 'width', 'height',
'align', 'style'],
  'td,th': ['class', 'align', 'valign', 'width', 'height', 'colspan', 'rowspan',
'bgcolor', 'style'],
  a : ['class', 'href', 'target', 'name', 'style'],
  embed : ['src', 'width', 'height', 'type', 'loop', 'autostart', 'quality',
'style', 'align', 'allowscriptaccess', '/'],
  img : ['src', 'width', 'height', 'border', 'alt', 'title', 'align', 'style',
'/'],
  hr : ['class', '/'],
  br : ['/'],
  'p,ol,ul,li,blockquote,h1,h2,h3,h4,h5,h6' : ['align', 'style'],
```

```
        'tbody,tr,strong,b,sub,sup,em,i,u,strike' : []  
    }  
}
```

注：filterMode 为 true 时有效。3.4 版本开始属性可设置 style，保留所有 inline 样式。

9. **resizeMode**

2 或 1 或 0，2 时可以拖动改变宽度和高度，1 时只能改变高度，0 时不能拖动。

数据类型：Int

默认值：2

10. **skinType**

风格类型，default 或 tinymce。

数据类型：String

默认值：default

注：3.4 以前版本和 3.4.1 版本开始默认值为 default，3.4 版本的默认值为 oxygen。

11. **wyswygMode**

可视化模式或代码模式

数据类型：Boolean

默认值：true

12. **cssPath**

指定编辑器 iframe document 的 CSS，用于设置可视化区域的样式。

数据类型：String 或 Array

默认值：空

注：3.4.1 版本开始可指定多个 CSS 文件。例如：cssPath : ['a.css', 'b.css']

13. **skinsPath**

指定编辑器的 skins 目录，skins 目录存放风格的 css 文件和 gif 图片。

数据类型：String

默认值：KE.scriptPath + 'skins/'

14. **pluginsPath**

指定编辑器的 plugins 目录。

数据类型：String

默认值：KE.scriptPath + 'plugins/'

15. **minChangeSize**

undo/redo 文字输入最小变化长度，当输入的文字变化小于这个长度时不会添加到 undo 堆栈里。

数据类型：Int

默认值：5

16. **loadStyleMode**

true 时自动加载编辑器的 CSS。

数据类型：Boolean

默认值：true

注：3.4 版本开始支持。

17. **urlType**

改变站内本地 URL，可设置空、relative、absolute、domain。空为不修改 URL，relative 为相对路径，absolute 为绝对路径，domain 为带域名的绝对路径。

数据类型：String

默认值：空

注：3.4 版本开始支持，3.4.1 版本开始默认值为空。

18. **newlineTag**

设置回车换行标签，可设置 p、br。

数据类型：String

默认值：br

注：3.4 版本开始支持。

19. **afterCreate**

设置编辑器创建后执行的回调函数。

数据类型：Function

默认值：无

20. **afterDialogCreate**

设置弹出浮动框创建后执行的回调函数。

数据类型：Function

默认值：无

注：3.4.3 版本开始支持。

21. **allowUpload**

true 或 false，true 时显示上传图片标签。

数据类型：Boolean

默认值：true

注：3.4 版本开始支持。

22. **allowFileManager**

true 或 false，true 时显示浏览服务器图片功能。

数据类型：Boolean

默认值：false

注：3.4 版本开始支持。

23. **referMethod**

设置 referMethod 后上传图片的 POST 参数里有 referMethod。

数据类型：String

默认值：空

注：3.4 版本开始支持。

24. **dialogAlignType**

设置弹出框(dialog)的对齐类型，可设置 page 和空，指定 page 时按当前页面居中，指定空时按编辑器居中。

数据类型：String

默认值：page

注：3.4.1 版本开始支持。

25. **imageUploadJson**

指定上传图片的服务器端程序。

数据类型：String

默认值：.././php/upload\_json.php

注：3.4.1 版本开始支持。

26. **fileManagerJson**

指定浏览远程图片的服务器端程序。

数据类型：String

默认值: ../../php/file\_manager\_json.php

注: 3.4.1 版本开始支持。

## 27. **autoSetDataMode**

true 或 false, true 时自动将编辑器内容设置到原来的 textarea, 也就是每次输入内容就执行 KE.util.setData 函数。

数据类型: Boolean

默认值: true

注: 3.5 版本开始支持。打开 autoSetDataMode 模式后会降低输入性能, 理想的做法是关闭 autoSetDataMode, 提交数据前执行 KE.util.setData。

## 28. **afterSetData**

自动执行 KE.util.setData 后执行的回调函数, autoSetDataMode 为 true 时有效。

数据类型: Function

默认值: 无

注: 3.5 版本开始支持。

## 29. **shadowMode**

true 或 false, true 时弹出层显示阴影。

数据类型: Boolean

默认值: true

注: 3.5 版本开始支持。

## 30. **allowPreviewEmoticons**

true 或 false, true 时鼠标放在表情上可以预览表情。

数据类型: Boolean

默认值: true

注: 3.5 版本开始支持。

参数设置例子:

```
KE.show({
  id : "content_1",
  width : "70%", //编辑器的宽度为 70%
  height : "200px", //编辑器的高度为 100px
  filterMode : false, //不会过滤 HTML 代码
  resizeMode : 1 //编辑器只能调整高度
});
```

### 一、添加"你好"插件

1. 定义 KE.lang['hello'] = "你好"。

```
KE.lang['hello'] = "您好";
```

2. 定义 KE.plugin['hello'], 所有逻辑都在这个对象里, 点击图标时默认执行 click 方法。

```

KE.plugin['hello'] = {
  click : function(id) {
    alert("您好");
  }
};

```

3. 页面里添加图标定义 CSS。

```

.ke-icon-hello {
  background-image: url(./skins/default.gif);
  background-position: 0px -672px;
  width: 16px;
  height: 16px;
}

```

4. 最后调用编辑器时 items 数组里添加 hello。

```

KE.show({
  id : 'content1',
  items : ['hello']
});

```

演示地址: [在新窗口打开](#)

## 二、添加插入远程图片的插件

1. 定义 KE.lang['remote\_image'] = "插入远程图片"。

```

KE.lang['remote_image'] = "插入远程图片";

```

2. 定义 KE.plugin['remote\_image']。

```

KE.plugin['remote_image'] = {
  click : function(id) {
    KE.util.selection(id);
    var dialog = new KE.dialog({
      id : id,
      cmd : 'remote_image',
      width : 310,
      height : 90,
      title : KE.lang['image'],
      yesButton : KE.lang['yes'],

```

```

        noButton : KE.lang['no']
    });
    dialog.show();
},
check : function(id) {
    var dialogDoc = KE.util.getIframeDoc(KE.g[id].dialog);
    var url = KE.$('url', dialogDoc).value;
    var title = KE.$('imgTitle', dialogDoc).value;
    var width = KE.$('imgWidth', dialogDoc).value;
    var height = KE.$('imgHeight', dialogDoc).value;
    var border = KE.$('imgBorder', dialogDoc).value;
    if (url.match(/\.(jpg|jpeg|gif|bmp|png)$/i) == null) {
        alert(KE.lang['invalidImg']);
        window.focus();
        KE.g[id].yesButton.focus();
        return false;
    }
    if (width.match(/^\d+$/i) == null) {
        alert(KE.lang['invalidWidth']);
        window.focus();
        KE.g[id].yesButton.focus();
        return false;
    }
    if (height.match(/^\d+$/i) == null) {
        alert(KE.lang['invalidHeight']);
        window.focus();
        KE.g[id].yesButton.focus();
        return false;
    }
    if (border.match(/^\d+$/i) == null) {
        alert(KE.lang['invalidBorder']);
        window.focus();
        KE.g[id].yesButton.focus();
        return false;
    }
    return true;
},
exec : function(id) {
    KE.util.select(id);
    var iframeDoc = KE.g[id].iframeDoc;
    var dialogDoc = KE.util.getIframeDoc(KE.g[id].dialog);
    if (!this.check(id)) return false;
    var url = KE.$('url', dialogDoc).value;
    var title = KE.$('imgTitle', dialogDoc).value;
    var width = KE.$('imgWidth', dialogDoc).value;

```

```

        var height = KE.$('imgHeight', dialogDoc).value;
        var border = KE.$('imgBorder', dialogDoc).value;
        this.insert(id, url, title, width, height, border);
    },
    insert : function(id, url, title, width, height, border) {
        var html = ' 0) html += 'width="' + width + '" ';
        if (height > 0) html += 'height="' + height + '" ';
        if (title) html += 'title="' + title + '" ';
        html += 'alt="' + title + '" ';
        html += 'border="' + border + '" />';
        KE.util.insertHtml(id, html);
        KE.layout.hide(id);
        KE.util.focus(id);
    }
};

```

3. 页面里添加图标定义 CSS。

```

.ke-icon-remote_image {
    background-image: url(./skins/default.gif);
    background-position: 0px -496px;
    width: 16px;
    height: 16px;
}

```

4. 最后调用编辑器时 items 数组里添加 remote\_image。

```

KE.show({
    id : 'content1',
    items : ['remote_image']
});

```

## API 文档

### 一、变量

1. **KE.scriptPath**  
kindeditor.js 的路径。  
数据类型: String
2. **KE.browser**  
浏览器类型和版本, 分别为 KE.browser.VERSION、KE.browser.IE、KE.browser.WEBKIT、KE.browser.GECKO、KE.browser.OPERA。



数据类型: Object

注: 3.4 以前版本直接返回字符串, 分别为"IE"、"WEBKIT"、"GECKO"、"OPERA"。

### 3. **KE.g[id].iframeDoc**

编辑区域的 iframe 对象。

数据类型: Element

### 4. **KE.g[id].keSel**

当前选中信息的 KE.selection 对象, id 为编辑器 ID。

数据类型: Object

### 5. **KE.g[id].keRange**

当前选中信息的 KE.range 对象, id 为编辑器 ID。

数据类型: Object

### 6. **KE.g[id].sel**

当前选中信息的浏览器原生 selection 对象, id 为编辑器 ID。

数据类型: Object

### 7. **KE.g[id].range**

当前选中信息的浏览器原生 range 对象, id 为编辑器 ID。

数据类型: Object

## 二、函数

### 1. **KE.show(config)**

初始化并创建编辑器。执行本函数时先调用 KE.init 设置初始化参数, 然后在 DOM 加载完成后执行 KE.create。

参数:

config: Object, 编辑器属性集合, 具体请参考编辑器属性

返回值: 无

### 2. **KE.init(config)**

设置编辑器的初始化参数, loadStyleMode 为 true 时自动加载 CSS 文件。

参数:

config: Object, 编辑器属性集合, 具体请参考编辑器属性

返回值: 无

### 3. **KE.create(id, mode)**

创建编辑器。

参数:

id: String, 编辑器的 ID

mode: Int, 可选参数, 指定 1 时在 body 下面创建编辑器, 0 或未指定时在 TEXTAREA 前面创建编辑器。

返回值: 无

### 4. **KE.remove(id, mode)**

移除编辑器。

参数:

id: String, 编辑器的 ID

mode: Int, 可选参数, 指定 1 时移除在 body 下面的编辑器, 0 或未指定时移除在 TEXTAREA 前面的编辑器。

返回值: 无

### 5. **KE.html(id)**

取得编辑器的 HTML 内容。

参数:

id: String, 编辑器的 ID

返回值:

String, HTML 字符串

注: 3.5 版本开始支持, 旧版本可以使用 `KE.util.getData(id)`。

6. **KE.html(id, val)**

设置编辑器的 HTML 内容。

参数:

id: String, 编辑器的 ID

val: String, HTML 字符串

返回值: 无

注: 3.5 版本开始支持, 旧版本可以使用 `KE.util.setFullHtml(id, val)`。

7. **KE.text(id)**

取得编辑器的纯文本内容。

参数:

id: String, 编辑器的 ID

返回值:

String, 去除 HTML 代码后的文本

注: 3.5 版本开始支持。

8. **KE.text(id, val)**

设置编辑器的内容, 直接显示 HTML 代码。

参数:

id: String, 编辑器的 ID

val: String, 文本

返回值: 无

注: 3.5 版本开始支持。

9. **KE.selectedHtml(id)**

取得当前被选中的 HTML 内容。

参数:

id: String, 编辑器的 ID

返回值:

String, HTML 字符串

注: 3.5 版本开始支持。

10. **KE.count(id, mode)**

取得编辑器内容的文字数量。

参数:

id: String, 编辑器的 ID

mode: String, 可选参数, 默认值也"html", mode 为"html"时取得字数包含 HTML 代码, mode 为"text"时只包含纯文本、IMG、EMBED。

返回值:

Int, 文字数量

注: 3.5 版本开始支持。

11. **KE.isEmpty(id)**

判断编辑器是否有可见内容, 比如文本、图片、视频。

参数:

id: String, 编辑器的 ID

返回值: Boolean

注: 3.5 版本开始支持, 旧版本可以使用 `KE.util.isEmpty(id)`。

#### 12. **KE.insertHtml(id, val)**

将指定的 HTML 内容插入到编辑区域里的光标处。

参数:

id: String, 编辑器的 ID

val: String, HTML 字符串

返回值: 无

注: 3.5 版本开始支持, 旧版本可以使用 `KE.util.insertHtml(id, val)`。

#### 13. **KE.appendHtml(id, val)**

将指定的 HTML 内容添加到编辑区域最后位置。

参数:

id: String, 编辑器的 ID

val: String, HTML 字符串

返回值: 无

注: 3.5 版本开始支持。

#### 14. **KE.util.setData(id)**

把编辑器的内容设置到原 TEXTAREA 控件里。

参数:

id: String, 编辑器的 ID

返回值: 无

## 常见问题

### 常见问题

1. 编辑器好像是 UTF-8 编码的, 可以在 GB2312 页面上使用吗?
2. 我想通过 ajax 方式加载编辑器, 但是加 `KE.show` 后没有效果。
3. 为什么会丢失某些 HTML 标签, 是不是编辑器的 BUG?
4. 调用编辑器后显示不正常, 怎么调整才能不和我的页面的 CSS 冲突?
5. 跨域调用后, 点击插入图片或超级链接会报错。
6. 我取不到编辑器数据, 直接取得 `textarea` 的 `value` 也没用。
7. 上传图片为什么出现“服务器发生故障。”?

#### 1. 编辑器好像是 UTF-8 编码的, 可以在 GB2312 页面上使用吗?

可以使用。有两种方法, 一种方法是引入 `kindeditor.js` 文件时将 `script` 的 `charset` 属性设置成 `utf-8`。还有一种方法是直接将 `html/js/css` 文件编码都转换成 GB2312 编码 (用 Notepad++、editPlus 等文本编辑器就可以转换编码), 不过转换格式后升级比较困难, 建议使用第一种方法。

```
<script type="text/javascript" charset="utf-8" src="/editor/kindeditor.js"></script>
```

#### 2. 我想通过 ajax 方式加载编辑器, 但是加 `KE.show` 后没有效果。

这个需要理解 `KE.show` 和 `KE.create` 的区别。`KE.show` 的原理是先执行 `KE.init` 设置一些变量, 等 DOM 全部创建以后才开始执行 `KE.create` 创建编辑器。如果浏览器不触发 `DOMContentLoaded` 事件 (例如: 点击某个按钮, 通过 `innerHTML` 插入 HTML 等), 则不能使用 `KE.show`, 需要直接调用 `KE.init` 和 `KE.create`。

还有我发现很多同学直接把 javascript 代码放在 ajax 返回的 HTML 页面里执行，这么做可能出现不可预测的错误，最好的方法是 ajax 只返回纯 HTML，返回成功后在该 ajax 方法的回调函数（callback function）里执行 javascript。

```
//ajax 请求之前执行 KE.init
KE.init({
    id : 'textarea_id'
});
//ajax 返回 HTML 后执行 KE.create 创建编辑器
KE.create('textarea_id');
```

### 3. 为什么会丢失某些 HTML 标签，是不是编辑器的 BUG?

这是因为 3.4 版本以前编辑器默认开启了过滤模式（filterMode:true）。当 filterMode 为 true 时，编辑器会根据 htmlTags 设定自动过滤 HTML 代码，主要是为了生成干净的代码。

如果想保留所有 HTML，请将 filterMode 设置成 false。如果想保留特定 HTML，请将 filterMode 设置成 true 后，配置 htmlTags 属性。

从 3.4 版本开始默认关闭过滤模式，所以要过滤 HTML，需要将 filterMode 设置成 true。

```
KE.show({
    id : 'textarea_id',
    filterMode : true // true:开启过滤模式, false:关闭过滤模式
});
```

### 4. 调用编辑器后显示不正常，怎么调整才能不和我的页面的 CSS 冲突?

KindEditor 所有 CSS 选择器都用 ke 开头的名字，所以名字是不会直接冲突的，一般页面显示不正常的原因是你的页面使用了全局标签的 CSS 定义。

解决方式是编辑器的 div、table、td 等标签重新设置一遍 CSS。首先将 skins/oxygen.css 文件放在你的 CSS 文件后面，然后在 oxygen.css 文件后面再定义一遍发生冲突的 CSS。

```
<style>
/* 你的全局 CSS 定义 */
img {
    border: 1px solid #555555;
    padding: 10px;
}
</style>

<!-- 编辑器的 CSS 文件 -->
<link rel="stylesheet" type="text/css" href="./skins/oxygen.css" />

<style>
/* 重新设置全局 CSS 定义 */
img {
    border: 0;
    padding: 0;
}
</style>

<!-- 显式包含 CSS 时，需要将 loadStyleMode 属性设置成 false -->
```

```
<script>
KE.show({
    id : 'textarea_id',
    loadStyleMode : false
});
</script>
```

##### 5. 跨域调用后，点击插入图片或超级链接会报错。

KindEditor 的弹出框是用 iframe 加载 HTML 文件后显示的，跨域后 2 个 document 的域名不一样，无法进行跨页面 js 调用。

有一个解决跨域问题的方法是 postMessage，可惜 IE6 和 IE7 都不支持 postMessage，需要用 VBScript 解决，具体实现可参考 [shindig](#) 代码。

还有一个方法是不用 iframe 加载外部 HTML 的模式，重新实现弹出框（KE.dialog）的代码。

##### 6. 我取不到编辑器数据，直接取得 textarea 的 value 也没用。

KindEditor 的可视化操作在新创建的 iframe 上执行，代码模式下的 textarea 框也是新创建的，所以最后提交前需要将 HTML 数据设置到原来的 textarea，KE.util.setData 函数会完成这个动作。KindEditor 在默认情况下自动寻找 textarea 所属的 form 元素，找到 form 后 onsubmit 事件里添加 KE.util.setData() 函数，所以您用传统方法使用编辑器，不需要额外执行 KE.util.setData 函数。

//将编辑器的 HTML 数据设置到 textarea

```
KE.util.setData(id);
```

//KE.util.setData 等价于以下代码

```
KE.g[id].srcTextarea.value = KE.util.getData(id);
```

##### 7. 上传图片为什么出现"服务器发生故障。"？

出现这个提示的根本原因是没有正确返回 JSON 数据。具体原因有很多，imageUploadJson 参数设置错误，upload\_json.xxx 程序出错都会出现这个提示。

这时用 Firefox 的 Firebug 插件调试比较方便，上传图片后在 Firebug 的"网络" > "HTML"里能看到"POST upload\_json.xxx"，正常情况下 HTTP 状态为 200，并返回正确的 JSON 格式，如图所示。

