

配置参数

名称	类型	默认值	描述
内容			
title	String	'消息'	标题内容
content	String	'loading..'	消息内容。 1、如果传入的是HTMLElement类型，如果是隐藏元素会为其设置display:block以显示该元素，其他属性与绑定的事件都会完整保留，对话框关闭后此元素又将恢复原来的display属性，并且重新插入原文档所在位置 2、如果没有设定content的值则会有loading的动画
	HTMLElement		
按钮			
ok	Function	null	确定按钮回调函数。 函数如果返回false将阻止对话框关闭；函数this指针指向内部api；如果传入true表示只显示有关闭功能的按钮
	Boolean		
cancel	Function	null	取消按钮回调函数。 函数如果返回false将阻止对话框关闭；函数this指针指向内部api；如果传入true表示只显示有关闭功能的按钮 对话框标题栏的关闭按钮其实就是取消按钮，只不过视觉不同罢了，点击同样触发cancel事件
	Boolean		
okVal	String	'确定'	"确定按钮"文字
cancelVal	String	'取消'	"取消按钮"文字
button	Array	null	自定义按钮。 配置参数成员： name —— 按钮名称 callback —— 按下后执行的函数 focus —— 是否聚焦点 disabled —— 是否标记按钮为不可用状态（后续可使用扩展方法让其恢复可用状态） 示例： 参数如：[{name: '登录', callback: function () {}}, {name: '取消'}]。 注意点击按钮默认会触发按钮关闭动作，需要阻止触发关闭请让回调函数返回false
尺寸			
width	Number	'auto'	设置消息内容宽度，可以带单位。一般不需要设置此，对话框框架会自己适应内容。 如果设置为百分值单位，将会以根据浏览器可视范围作为基准，此时如果浏览器窗口大小被改变其也会进行相应的调整
	String		
height	Number	'auto'	设置消息内容高度，可以带单位。不建议设置此，而应该让内容自己撑开高度。 如果设置为百分值单位，将会以根据浏览器可视范围作为基准，此时如果浏览器窗口大小被改变其也会进行相应的调整
	String		
位置			
fixed	Boolean	false	开启静止定位。静止定位是css2.1的一个属性，它静止在浏览器某个地方不动，也不受滚动条拖动影响。（artDialog支持IE6 fixed）

follow	HTMLElement	null	让对话框依附在指定元素附近。
	String		可传入元素ID名称，注意ID名称需要以 “#” 号作为前缀
left	Number	'50%'	相对于可视区域的X轴的坐标。
	String		可以使用'0%' ~ '100%'作为相对坐标，如果浏览器窗口大小被改变其也会进行相应的调整
top	Number	'38.2%'	相对于可视区域的Y轴的坐标。
	String		可以使用'0%' ~ '100%'作为相对坐标，如果浏览器窗口大小被改变其也会进行相应的调整
视觉			
lock	Boolean	false	开启锁屏。 中断用户对话框之外的交互，用于显示非常重要的操作/消息，所以不建议频繁使用它，它会让操作变得繁琐
background	String	'#000'	锁屏遮罩颜色
opacity	Number	0.7	锁屏遮罩透明度
icon	String	null	定义消息图标。可定义 “skins/icons/” 目录下的图标名作为参数名（不包含后缀名）
padding	String	'20px 25px'	内容与边界填充边距(即css padding)
交互			
time	Number	null	设置对话框显示时间。以秒为单位
resize	Boolean	true	是否允许用户调节尺寸
drag	Boolean	true	是否允许用户拖动位置
esc	Boolean	true	是否允许用户按Esc键关闭对话框
高级			
id	String	null	设定对话框唯一标识。用途：
	Number		1、防止重复弹出 2、定义id后可以使用art.dialog.list[youID]获取扩展方法
zIndex	Number	1987	重置全局zIndex初始值。 用来改变对话框叠加高度。比如有时候配合外部浮动层UI组件，但是它们可能默认zIndex没有artDialog高，导致无法浮动到artDialog之上，这个时候你就可以给对话框指定一个较小的zIndex值。 请注意这是一个会影响到全局的配置，后续出现的对话框叠加高度将重新按此累加。
init	Function	null	对话框弹出后执行的函数
close	Function	null	对话框关闭前执行的函数。函数如果返回false将阻止对话框关闭。 请注意这不是关闭按钮的回调函数，无论何种方式关闭对话框，close都将执行。
show	Boolean	true	是否显示对话框

扩展方法

名称	描述
close()	关闭对话框
show()	显示对话框
hide()	隐藏对话框
title(value)	写入标题。无参数则返回标题容器元素
content(value)	向消息容器中写入内容。参数支持字符串、DOM对象，无参数则返回内容容器元素
	插入一个自定义按钮。

button(arguments)	<p>配置参数成员：</p> <p>name —— 按钮名称</p> <p>callback —— 按下后执行的函数</p> <p>focus —— 是否聚焦点</p> <p>disabled —— 是否标记按钮为不可用状态（后续可使用扩展方法让其恢复可用状态）</p> <p>示例：</p> <pre>button({ name: '登录', focus: true, callback: function () {} }, { name: '取消' })</pre>
follow(element)	让对话框吸附到指定元素附近
position(left, top)	重新定位对话框
size(width, height)	重新设定对话框大小
lock()	锁屏
unlock()	解锁
time(val)	定时关闭（单位秒）

配置参数演示

内容 [content]

1. 传入字符串

```
art.dialog({
  content: '我支持HTML'
});
```

运行»

2. 传入HTMLElement

备注：1、元素不是复制而是完整移动到对话框中，所以原有的事件与属性都将会保留 2、如果隐藏元素被传入到对话框，会设置display:block属性显示该元素 3、对话框关闭的时候元素将恢复到原来在页面的位置，style display属性也将恢复

```
art.dialog({
  content: document.getElementById('demoCode_content_DOM'),
  id: 'EF893L'
});
```

运行»

标题 [title]

```
art.dialog({
  title: 'hello world!'
});
```

运行»

确定取消按钮 [ok & cancel]

备注：回调函数this指向扩展接口，如果返回false将阻止对话框关闭

```
art.dialog({
  content: '如果定义了回调函数才会出现相应的按钮',
  ok: function () {
    this.title('3秒后自动关闭').time(3);
    return false;
  },
  cancelVal: '关闭',
  cancel: true //为true等价于function() {}
});
```

运行»

自定义按钮 [button]

备注：回调函数this指向扩展接口，如果返回false将阻止对话框关闭；button参数对应的扩展方法名称也是"button"

```
art.dialog({
  id: 'testID',
  content: 'hello world!',
  button: [
    {
      name: '同意',
      callback: function () {
        this.content('你同意了').time(2);
        return false;
      },
      focus: true
    },
    {
      name: '不同意',
      callback: function () {
        alert('你不同意')
      }
    },
    {
      name: '无效按钮',
      disabled: true
    },
    {
      name: '关闭我'
    }
  ]
});
```

运行»

定时关闭的消息 [time]

```
art.dialog({
  time: 2,
  content: '两秒后关闭'
});
```

运行»

防止重复弹出 [id]

```
art.dialog({
  id: 'testID2',
  content: '再次点击运行看看'
});
art.dialog({id: 'testID2'}).title('3秒后关闭').time(3);
```

运行»

定义消息图标 [icon]

请查看skins/icons目录下的图标

```
art.dialog({
    icon: 'succeed',
    content: '我可以定义消息图标哦'
});
```

运行»

内容与边界填充边距 [padding]

有时候并不需要默认的内容填充边距，如展示图片与视频

```
art.dialog({
    padding: 0,
    title: '照片',
    content: '',
    lock: true
});
```

运行»

锁屏 [lock & background & opacity]

```
art.dialog({
    lock: true,
    background: '#600', // 背景色
    opacity: 0.87, // 透明度
    content: '中断用户在对话框以外的交互，展示重要操作与消息',
    icon: 'error',
    ok: function () {
        art.dialog({content: '再来一个锁屏', lock: true});
        return false;
    },
    cancel: true
});
```

运行»

跟随元素 [follow]

1、标准形式

```
art.dialog({
    follow: document.getElementById('followTestBtn'),
    content: '让对话框跟着某个元素，一个元素同时只能出现一个对话框'
});
```

运行»

2、使用简写形式（已经绑定onclick事件，注意此返回值不再是对话框扩展方法）

```
art('#demoCode_follow_a').dialog({
    content: '让对话框跟着某个元素，一个元素同时只能出现一个对话框'
});
```

[元素触发](#)

运行»

自定义坐标 [left & top]

```
art.dialog({
    left: 100,
    top: '60%',
    content: '我改变坐标了'
});
```

运行»

创建一个右下角浮动的消息窗口

```
art.dialog({
  id: 'msg',
  title: '公告',
  content: 'hello world!',
  width: 320,
  height: 240,
  left: '100%',
  top: '100%',
  fixed: true,
  drag: false,
  resize: false
})
```

[运行»](#)

设置大小 [width & height]

```
art.dialog({
  width: '20em',
  height: 55,
  content: '尺寸可以带单位'
});
```

[运行»](#)

创建一个全屏对话框

```
art.dialog({
  width: '100%',
  height: '100%',
  left: '0%',
  top: '0%',
  fixed: true,
  resize: false,
  drag: false
})
```

[运行»](#)

静止定位 [fixed]

```
art.dialog({
  fixed: true,
  content: '请拖动滚动条查看'
});
```

[运行»](#)

不许拖拽 [drag & resize]

```
art.dialog({
  drag: false,
  resize: false,
  content: '禁止拖拽'
});
```

[运行»](#)

扩展方法演示

特别说明：扩展方法支持链式操作

获取扩展方法一：直接引用返回

```
var dialog = art.dialog({
```

```
        title: '我是对话框',
        content: '我是初始化的内容'
    });

    dialog.content('对话框内容被扩展方法改变了').title('提示');
```

运行»

获取扩展方法二：通过对话框ID

```
art.dialog({
    id: 'Kdf435',
    title: '警告',
    content: '我是初始化的内容'
});

art.dialog.list['Kdf435'].content('对话框内容被扩展方法改变了');
```

运行»

获取扩展方法三：回调函数中this

```
art.dialog({
    title: '警告',
    content: '我是初始化的内容',
    ok: function () {
        this.content('你点了确定按钮').lock();
        return false;
    },
    init: function () {
        this.content('对话框内容被扩展方法改变了');
    }
});
```

运行»

关闭页面所有对话框

```
var list = art.dialog.list;
for (var i in list) {
    list[i].close();
};
```

运行»

按钮接口演示

备注：回调函数如果返回false将阻止对话框关闭

```
var dialog = art.dialog({
    title: '警告',
    content: '点击管理按钮将让删除按钮可用',
    width: '20em',
    button: [{
        name: '管理',
        callback: function () {
            this
                .content('我更改了删除按钮')
                .button({
                    name: '删除',
                    disabled: false
                })
                .lock();
            return false;
        },
        focus: true
    }]
});
```

```
dialog.button({
  {
    name: '删除',
    callback: function () {
      alert('delect')
    },
    disabled: true
  }
})
```

运行»

标题倒计时

```
var timer;
art.dialog({
  content: '时间越来越少，越来越少..',
  init: function () {
    var that = this, i = 5;
    var fn = function () {
      that.title(i + '秒后关闭');
      !i && that.close();
      i --;
    };
    timer = setInterval(fn, 1000);
    fn();
  },
  close: function () {
    clearInterval(timer);
  }
}).show();
```

运行»

关闭不删除内容

```
art.dialog({
  id: 'show-hide',
  content: '关闭后阻止对话框被删除，只隐藏对话框',
  close: function () {
    this.hide();
    return false;
  }
}).show();
```

运行»

AJAX高级应用：执行HTML片段中特殊script标签

HTML片段中的 `<script type="text/dialog"></script>` 标签包裹的javascript将会在artDialog内部执行，其this指向对话框扩展方法，这样可以进一步实现消息内容模块化。

(本例子使用了AJAX，需要在服务器上运行。可打开 [./ajaxContent/login.html](#) 查看源码中的自定义脚本)

```
var dialog = art.dialog({id: 'N3690',title: false});

// jQuery ajax
$.ajax({
  url: './ajaxContent/login.html',
  success: function (data) {
    dialog.content(data);
  },
  cache: false
});
```

运行»

插件：iframe应用工具

请新打开新页面查看范例：[./iframeTop.html](#)

应用到你的项目

在页面head引入artDialog，其中skin参数表示对话框使用的风格文件(如果项目采用jQuery作为框架，则引用jQuery artDialog版本)。

```
<script src="artDialog/artDialog.js?skin=default"></script>
```

如果需要在iframe应用下提供 [更好的支持](#)，可加载插件：

```
<script src="artDialog/plugins/iframeTools.js"></script>
```

配置全局默认参数(可选):

```
(function (config) {  
    config['lock'] = true;  
    config['fixed'] = true;  
    config['okVal'] = 'Ok';  
    config['cancelVal'] = 'Cancel';  
    // [more...]  
})(art.dialog.defaults);
```