基础参数

| 键: 值 | 描述 |
| --- | --- |
| 下表的属性都是默认值，您可在调用时按需重新配置，他们可帮助你实现各式各样的风格。如是调用： $.layer({键: 值, 键: 值, …}); | |
| type: 0 | 层的类型。0：信息框（默认），1：页面层，2：iframe层，3：加载层，4：tips层。  此为重要参数，不同类型层的总开关，若为type:0则不需要配置，其它类型层在调用时必须设置type。 |
| title: '信息' | 控制默认标题栏。 如不想显示标题栏，配置title: false 即可 |
| maxmin: false | 是否输出窗口最小化/全屏/还原按钮。  如需要开启，设置maxmin: true 即可  此功能为layer1.8开始新增 |
| offset: ['220px' , ''] | 控制层坐标。 offset的值分别是： [纵坐标, 横坐标]，横坐标为空表示水平居中 如果您想垂直居中，您可以offset: [($(window).height() - 层的高度值)/2 + 'px', ''] |
| area: ['310px', '130px'] | 控制层宽高。 area值分别为：[宽度, 高度] 当设置为auto时，意味着采用自适应（iframe层不能设置auto）, 对于宽度，并不推荐您设置auto。 |
| border: [10, 0.3, '#000'] | 控制层的边框。 border的值分别为：[边框大小, 透明度, 颜色, layer1.8之前需在此处加true] 如果您不想显示border，设置 border: [0] 即可 |
| shade: [0.5, '#000'] | 控制遮罩。 值分别是：[遮罩透明度, 遮罩颜色, layer1.8之前需在此处加true]  如果不想显示遮罩，配置shade: [0]即可 |
| shadeClose: false | 用来控制点击遮罩区域是否关闭层。 若开启，设为true即可 |
| closeBtn: [0, true] | 控制层右上角关闭按钮。 closeBtn的值分别为: [关闭按钮的风格（支持0和1）, true] 若不想显示关闭按钮，配置 closeBtn: false即可 |
| time: 0 | 自动关闭等待秒数，整数值。 0表示不自动关闭，若3秒后自动关闭，type: 3即可 |
| fix: true, | 用于设定层是否不随滚动条而滚动，固定在可视区域。 |
| move: '.xubox\_title' | 设定某个元素来实现对层的拖拽。 值为：用来拖拽的元素选择器 若不想拖拽，move: false即可 |
| moveOut: false | 用于控制层是否允许被拖出可视窗口外 |
| moveType: 0 | 用于配置拖拽类型（layer1.7之前版本不支持） 默认为引导式拖动层，若值设为1，则直接拖动层 |
| bgcolor: '#fff' | 用于控制层的背景色 如果不想设置任何颜色，设置空字符即可。但是对于type:0的对话框层而言，始终都是白色 |
| zIndex: 19891014 | 控制层堆叠顺序（即css的z-index）。整数值。 合理设置它，可以避免与其它插件的层级冲突 |
| maxWidth: 400 | 最大宽度。整数值。 当area宽度设为auto时才有用。 |
| fadeIn: 300, | 用于控制层渐显弹出（layer1.7之前版本不支持） 值为毫秒数 |
| btns: 1, | 按钮的个数。提供了0-2的选择，设置0表示不输出按钮 |
| btn: ['确定', '取消'], | [按钮一的文本值 , 按钮二的文本值] 必须btns值大于0才有效 |
| dialog: {  type: 3,  msg: ''  } | 信息框层模式提供的私有参数。使用时，按需配置即可  type: 图标类型，提供了0-16的选择，也许有你喜欢的。 设置-1不显示图标  msg: 信息框内容，重要参数 |
| page: {  dom: '#id',  html: '',  url: '',  ok: function(datas){}  } | 页面层模式私有参数。  dom: 页面已存在的选择器  html: 直接传入的html字符串。  url: ajax请求地址。  ok: ajax请求完毕后执行的回调，datas是异步传递过来的值。  需要特别注意的是，dom、html、url只需设定其中一个就行，若配置html或url，你必须也配置宽高值。 |
| iframe: {  src: '',  scrolling: 'auto'  } | iframe层模式私有参数。  src: 要打开的网址。  scrolling: 是否允许iframe出现滚动条，默认自动。允许：'yes'，不允许：'no' |
| loading: {  type: 0  } | 加载层私有属性。  type: loading图标类型（提供了0-3的选择）。  一般配合ajax使用 |
| tips : {  msg: '',  follow: '#id',  guide: 0,  isGuide: true,  style: ['', '']  } | tips提示层私有属性。  msg: 提示内容。  follow: 吸附目标选择器。  guide: 指引方向（0：上，1：右，2：下，3：左）。  isGuide: 是否显示默认三角形。 这个参数可配合msg帮助你自定义三角形icon  style: ['background-color:#FFF8ED; color:#000; border:1px solid #FF9900; /\* 此处可用来自定义tips的css样式 \*/', '#FF9900']]。 数组第二个值，为三角形的颜色。 |
| 回调函数 | |
| success: function(layero){  } | 层弹出成功后的回调函数.  layero是回调传过来的当前层容器的实例，这意味着你可以对当前弹层进行dom操作 |
| yes: function(index){} | 按钮一的回调函数  index为当前层的索引，主要用来回调执行后，配合layer.close(index)来关闭层 |
| no: function(index){} | 按钮二的回调函数 |
| close: function(index){} | 层右上角关闭按钮的点击事件触发回调函数。 |
| end: function(){} | 层被彻底关闭后执行的回调函数。 |
| moveEnd: function(){} | 拖拽完毕触发的回调函数 |
| min: function(layero){},  full: function(layero){},  restore: function(layero){} | 分别为最小化、全屏、还原触发后的回调函数  layero是当前层容器的实例  layer1.8开始新增 |

集成方法：

| 方法名 | 描述 |
| --- | --- |
| $.layer({基础参数}) | 核心接口，参数是一个对象，对象属性参见基础参数。  诸如layer.alert/layer.confirm/layer.msg/layer.tips等皆为$.layer()的二次封装。 |
| layer.v | 获取layer版本号。 |
| layer.alert(msg, icon, fn) | 对单按钮信息框的重新封装  参数分别为: 提示内容, 图标类型(-1到16的选择), 回调函数或标题  如：layer.alert('你好layer', 9); |
| layer.confirm(msg, yes, no) | 对询问框的重新封装  参数分别为: 提示内容, 确定回调函数, 取消回调函数或标题  如：  layer.confirm('确定删除吗？', function(){  //删除成功  }); |
| layer.msg(msg, time, parme) | 对无标题栏信息框层的重新封装  参数分别为: 提示内容, 自动关闭秒数（默认2）, 对象或回调函数或图标类型  当parme是一个对象时，您可配置{type: 图标, shade: 是否开启遮罩(默认true), rate: 弹出的动画类型，具体值见layer.shift的参数说明}  当parme是一个函数时，则将作为层消失后的end的回调  当parme是一个数字时，则将作为msg的图标类型  如：  layer.msg('删除成功', 2, function(){  location.reload(); //自动关闭后可做一些刷新页面等操作  }); |
| layer.load(parme, loadStype) | 对加载层的重新封装  参数分别为：加载提示语或自动关闭所需秒数, 加载风格（0-3的选择）  如果parme是一段文字，您可以这样：layer.load('加载中');  如果parme是一个数字，则为自动关闭所需最长等待秒数(0时则不自动关闭) ，如：layer.load(3) //3秒后自动消失 |
| layer.tips(html, follow, parme) | 对tips层的重新封装。  参数分别为：tips内容, 要吸引的dom对象, 对象  parme允许传这些属性{time: 自动关闭所需秒, maxWidth: 最大宽度, guide: 指引方向, style: tips样式（参加api表格一中的style属性）}  如：  obj.on('click', function(){  layer.tips('这是小提示','#id');  }); |
| layer.close(index) | 用于手动关闭层。  index参数为调用layer时返回的索引值。如:  var index = $.layer({此处是您的配置}); //诸如 var index = layer.alert();之类的也会返回index索引  //关闭时，只要把index传给close方法即可关闭对应的层，如：  layer.close(index); |
| layer.closeTips() | 用于关闭tips层。 |
| layer.closeAll() | 用于关闭所有层，在多层模式时也许会用到 |
| layer.area(index, options) | 用于重定义指定层的宽高、坐标、z-index等。  参数分别为：层的索引, 对象  options允许传这些属性 {width:宽度, height: 高度, top: 纵坐标, left:横坐标, zIndex:z-index值}等css属性。  该方法仅针对页面层（type:1）和iframe层（type:2）有效; |
| layer.autoArea(index) | 用来处理在层中宽高改变时，重新自适应层宽高。必填参数为层的索引值 |
| layer.getChildFrame(selector, index) | 获取子iframe中的DOM。一般用于父窗口操作iframe页。  参数分别为：iframe页的选择器, 层索引 |
| layer.getFrameIndex(window.name) | 获取当前所在iframe层的索引。  只允许在iframe页面内部调用。如在内部关闭自身：  var index = parent.layer.getFrameIndex(window.name);  parent.layer.close(index); |
| layer.iframeAuto(index) | 用于让iframe层自适应。  index为层的索引 |
| layer.iframeSrc(index, url) | 用于重置iframe层的url 参数分别为: 层的索引, url链接 |
| layer.shift(type, rate, stop) | layer弹出时内置动画，在success回调中使用。  参数分别为：动画类型, 动画速率（毫秒）, 底部动画是否到offset设定好的位置  type有七种选择：左上(left-top),上(top), 右上(right-top),右下(right-bottom),下(bottom),左下(left-bottom),左('left')。  如：  success: function(){  layer.shift('right-bottom', 400); //右下角弹出，400毫秒速率  } |
| layer.setMove() | 当拖拽元素改变时，可用此方法重新初始化拖拽。 |
| layer.setTop(layerNow) | 引用此方法可开启点击使当前窗口置顶功能，只能用于用在success回调中。  layerNow参数即为success回调函数中传过来的参数，即当前layer容器。当需要时，需配置zIndex:layer.zIndex， 详见[场景实例]页面的例子：无限层中层 |
| layer.min(index, options) layer.full(index, options) layer.restore(index) | 三个方法功能分别为：最小化、全屏、还原  index为层的索引，可通过var index = $.layer({})得到  options需传入层{area: [正常尺寸下的宽度, 正常尺寸下的高度], offset: [正常尺寸下的top, 正常尺寸下的left]}  如：layer.full(1, {area: ['300px', '200px', offset: ['200px', '']]}); 即可自己去控制什么地方操作全屏 |
| layer.use(module, callback) | 载入一个模块，支持载入js、css。  参数分别为：css或js路径, 加载完毕回调函数  如载入一个js：layer.use('extend/layer.ext.js', function(){}); |
| layer.index | 静态属性，用于获取最近一次触发的层索引值 |
| layer.zIndex | 静态属性，用于获取layer容器中的最大z-index值 |
| layer.path | 静态属性，获取当前layer的存放路径。 |

拓展方法：

| 方法名 | 描述 |
| --- | --- |
| 目前拓展方法主要是指layer.ext.js提供的拓展类，您可以按照自己的需求选择引入，同样的，不需要引入css，只需要执行 layer.use('extend/layer.ext.js') 即可，关于layer.use的使用，请看集成方法的介绍; | |
| layer.ext = function(){} | 首次加载layer.ext.js模块完毕的回调方法。  如果页面一加载即执行拓展层，需用到此方法。  如：  layer.use('extend/layer.ext.js', function(){  layer.ext = function(){  layer.prompt();  }  }) |
| layer.prompt(options, yes, no); | 用于弹出一个prompt层模式  参数分别为：对象, 确定的回调函数, 取消的回调函数  options允许你配置：  {  top: 高度,  type: 0, //支持普通文本框（0）、密码框（1）、文件框（2）、多行文本框（3）,  title: '标题',  length: 200 //表单框可输入的最长值  }    如：  layer.prompt({title: '测试'}, function(val){  alert('得到：'+ val);  })    [试一试](javascript:new%20Function($('#layerPro').html())();) |
| layer.tab(parme) | 实现弹出tab层。  parme是一个对象，它允许传入{data: [], area: 宽高, offset: 坐标} 如：  layer.tab({  data:[  {title: '标题一', content:'内容一，可加html'},  {title: '标题二', content:'内容二。什么？嫌tab层太难看？噢是的，这只是个骨架。'}  ],  area: ['600px', '300px'] //宽度，高度  });    [试一试](javascript:new%20Function($('#layerTab').html())();) |
| layer.photos(options) | 实现图片的相册模式浏览。支持异步请求过来的json数据、支持直接弹出页面指定区域所有图片  options是一个对象，具体调用，[深入了解](javascript:;layer.msg('%E8%BF%98%E5%9C%A8%E6%95%B4%E7%90%86',2,9)) |
| layer.photosPage(options) | 直接弹出页面指定区域所有图片以实现相册浏览。他是对layer.photos的二次封装  options同样是一个对象，具体调用，[深入了解](javascript:;layer.msg('%E8%BF%98%E5%9C%A8%E6%95%B4%E7%90%86',2,9)) |