使用场景

|  |  |  |
| --- | --- | --- |
| 2. layui 模块化使用 | 如果你使用的是 layui，那么你直接在官网下载 layui 框架即可，无需引入 jQuery 和 layer.js，但需要引入*layui.css*和*layui.js* | 通过*layui.use('layer', callback)*加载模块 |

1. layui.use('layer', function(){
2. var layer = layui.layer;
4. layer.msg('hello');
5. });

基础参数

我们提到的基础参数主要指调用方法时用到的配置项，如：*layer.open({content: ''})layer.msg('', {time: 3})*等

type - 基本层类型

类型：Number，默认：0

可传入的值有：*0*（信息框，默认）*1*（页面层）*2*（iframe层）*3*（加载层）*4*（tips层）。 若你采用*layer.open({type: 1})*方式调用，则type为必填项（信息框除外）

title - 标题

title支持三种类型的值，若你传入的是普通的字符串，如*title :'我是标题'*，那么只会改变标题文本；若你还需要自定义标题区域样式，那么你可以*title: ['文本', 'font-size:18px;']*

content - 内容

content可传入的值是灵活多变的，不仅可以传入普通的html内容，还可以指定DOM，更可以随着type的不同而不同。譬如：

1. /!\*
2. 如果是页面层
3. \*/
4. layer.open({
5. type: 1,
6. content: '传入任意的文本或html' //这里content是一个普通的String
7. });
8. layer.open({
9. type: 1,
10. content: $('#id') //这里content是一个DOM，注意：最好该元素要存放在body最外层，否则可能被其它的相对元素所影响
11. });
12. //Ajax获取
13. $.post('url', {}, function(str){
14. layer.open({
15. type: 1,
16. content: str //注意，如果str是object，那么需要字符拼接。
17. });
18. });
19. /!\*
20. 如果是iframe层
21. \*/
22. layer.open({
23. type: 2,
24. content: 'http://sentsin.com' //这里content是一个URL，如果你不想让iframe出现滚动条，你还可以*content: ['http://sentsin.com', 'no']*
25. });
26. /!\*
27. 如果是用layer.open执行tips层
28. \*/
29. layer.open({
30. type: 4,
31. content: ['内容', '#id'] //数组第二项即吸附元素选择器或者DOM
32. });

skin - 样式类名

目前layer内置的skin有：*layui-layer-lanlayui-layer-molv*，未来我们还会选择性地内置更多，但更推荐您自己来定义。

1. //单个使用
2. layer.open({
3. skin: 'demo-class'
4. });
5. //全局使用。即所有弹出层都默认采用，但是单个配置skin的优先级更高
6. layer.config({
7. skin: 'demo-class'
8. })
9. //CSS
10. body .demo-class .layui-layer-title{background:#c00; color:#fff; border: none;}
11. body .demo-class .layui-layer-btn{border-top:1px solid #E9E7E7}
12. body .demo-class .layui-layer-btn a{background:#333;}
13. body .demo-class .layui-layer-btn .layui-layer-btn1{background:#999;}
14. …
15. 加上body是为了保证优先级。你可以借助Chrome调试工具，定义更多样式控制层更多的区域。

area - 宽高

*area: ['500px', '300px']*

btn - 按钮

类型：String/Array，默认：'确认'

当您只想自定义一个按钮时，你可以*btn: '我知道了'*，当你要定义两个按钮时，你可以*btn: ['yes', 'no']*。当然，你也可以定义更多按钮，比如：*btn: ['按钮1', '按钮2', '按钮3', …]*，按钮1的回调是yes，而从按钮2开始，则回调为btn2: function(){}

1. layer.open({
2. content: 'test'
3. ,btn: ['按钮一', '按钮二', '按钮三']
4. ,yes: function(index, layero){
5. //按钮【按钮一】的回调
6. }
7. ,btn2: function(index, layero){
8. //按钮【按钮二】的回调
9. //return false 开启该代码可禁止点击该按钮关闭
10. }
11. ,btn3: function(index, layero){
12. //按钮【按钮三】的回调
13. //return false 开启该代码可禁止点击该按钮关闭
14. }
15. ,cancel: function(){
16. //右上角关闭回调
17. //return false 开启该代码可禁止点击该按钮关闭
18. }
19. });

closeBtn - 关闭按钮

类型：String/Boolean，默认：1

layer提供了两种风格的关闭按钮，可通过配置*1*和*2*来展示，如果不显示，则*closeBtn: 0*

time - 自动关闭所需毫秒

类型：Number，默认：0

默认不会自动关闭。当你想自动关闭时，可以*time: 5000*，即代表5秒后自动关闭

anim - 弹出动画

类型：Number，默认：0

| 值 | 备注 |
| --- | --- |
| anim: 0 | 平滑放大。默认 |
| anim: 1 | 从上掉落 |
| anim: 2 | 从最底部往上滑入 |
| anim: 3 | 从左滑入 |
| anim: 4 | 从左翻滚 |
| anim: 5 | 渐显 |
| anim: 6 | 抖动 |

maxWidth - 最大宽度

类型：Number，默认：360

maxHeight - 最大高度

类型：Number，默认：无

zIndex - 层叠顺序

类型：，默认：19891014（贤心生日 0.0）

一般用于解决和其它组件的层叠冲突。

success - 层弹出后的成功回调方法

1. layer.open({
2. content: '测试回调',
3. success: function(layero, index){
4. console.log(layero, index);
5. }
6. });

yes - 确定按钮回调方法

yes - 确定按钮回调方法

类型：Function，默认：null

该回调携带两个参数，分别为当前层索引、当前层DOM对象。如：

1. layer.open({
2. content: '测试回调',
3. yes: function(index, layero){
4. //do something
5. layer.close(index); //如果设定了yes回调，需进行手工关闭
6. }
7. });

cancel - 右上角关闭按钮触发的回调

cancel - 右上角关闭按钮触发的回调

类型：Function，默认：null

该回调携带两个参数，分别为：当前层索引参数（index）、当前层的DOM对象（layero），默认会自动触发关闭。

end - 层销毁后触发的回调

类型：Function，默认：null

无论是确认还是取消，只要层被销毁了，end都会执行，不携带任何参数。

layer.ready(callback) - 初始化就绪

当你在页面一打开就要执行弹层时，你最好是将弹层放入ready方法中，如：

### code[layui.code](http://www.layui.com/doc/modules/code.html" \t "https://www.layui.com/doc/modules/_blank)

1. //页面一打开就执行弹层
2. layer.ready(function(){
3. layer.msg('很高兴一开场就见到你');
4. });

layer.open(options) - 原始核心方法

layer.alert(content, options, yes) - 普通信息框

1. //eg1
2. layer.alert('只想简单的提示');
3. //eg2
4. layer.alert('加了个图标', {icon: 1}); //这时如果你也还想执行yes回调，可以放在第三个参数中。
5. //eg3
6. layer.alert('有了回调', function(index){
7. //do something
9. layer.close(index);
10. });

layer.confirm(content, options, yes, cancel) - 询问框

layer.msg(content, options, end) - 提示框

layer.load(icon, options) - 加载层

load并不需要你传太多的参数，但如果你不喜欢默认的加载风格，你还有选择空间。icon支持传入*0-2*如果是0，无需传。另外特别注意一点：*load默认是不会自动关闭的*，因为你一般会在ajax回调体中关闭它。

1. //eg1
2. var index = layer.load();
3. //eg2
4. var index = layer.load(1); //换了种风格
5. //eg3
6. var index = layer.load(2, {time: 10\*1000}); //又换了种风格，并且设定最长等待10秒
7. //关闭
8. layer.close(index);

layer.close(index) - 关闭特定层

1. //当你想关闭当前页的某个层时
2. var index = layer.open();
3. var index = layer.alert();
4. var index = layer.load();
5. var index = layer.tips();
6. //正如你看到的，每一种弹层调用方式，都会返回一个index
7. layer.close(index); //此时你只需要把获得的index，轻轻地赋予layer.close即可

layer.closeAll(type) - 关闭所有层

1. layer.closeAll(); //疯狂模式，关闭所有层
2. layer.closeAll('dialog'); //关闭信息框
3. layer.closeAll('page'); //关闭所有页面层
4. layer.closeAll('iframe'); //关闭所有的iframe层
5. layer.closeAll('loading'); //关闭加载层
6. layer.closeAll('tips'); //关闭所有的tips层