全局配置

目前支持的全局配置项如下：

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

1. layui.config({
2. dir: '/res/layui/' //layui.js 所在路径（注意，如果是script单独引入layui.js，无需设定该参数。），一般情况下可以无视
3. ,version: false //一般用于更新模块缓存，默认不开启。设为true即让浏览器不缓存。也可以设为一个固定的值，如：201610
4. ,debug: false //用于开启调试模式，默认false，如果设为true，则JS模块的节点会保留在页面
5. ,base: '' //设定扩展的Layui模块的所在目录，一般用于外部模块扩展
6. });

定义模块

方法：*layui.define([mods], callback)*

通过该方法可定义一个 *Layui模块*。参数mods是可选的，用于声明该模块所依赖的模块。callback即为模块加载完毕的回调函数，它返回一个exports参数，用于输出该模块的接口。

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

1. layui.define(function(exports){
2. //do something
4. exports('demo', function(){
5. alert('Hello World!');
6. });
7. });

exports是一个函数，它接受两个参数，第一个参数为模块名，第二个参数为模块接口，当你声明了上述的一个模块后，你就可以在外部使用了，demo就会注册到layui对象下，即可通过 *layui.demo()* 去执行该模块的接口。

加载所需模块

方法：*layui.use([mods], callback)*

Layui的内置模块并非默认就加载的，他必须在你执行该方法后才会加载。它的参数跟上述的 define方法完全一样。   
另外请注意，mods里面必须是一个合法的模块名，不能包含目录。

1. layui.use(['laypage', 'layedit'], function(){
2. var laypage = layui.laypage
3. ,layedit = layui.layedit;
4. //do something
5. });

该方法的函数其实返回了所加载的模块接口，所以你其实也可以不通过layui对象赋值获得接口（这一点跟Sea.js很像哈）：

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

1. layui.use(['laypage', 'layedit'], function(laypage, layedit){
3. //使用分页
4. laypage();
6. //建立编辑器
7. layedit.build();
8. });

动态加载CSS

方法：*layui.link(href)*

href即为css路径。注意：该方法并非是你使用Layui所必须的，它一般只是用于动态加载你的外部CSS文件。

本地存储

本地存储是对 localStorage 和 sessionStorage 的友好封装，可更方便地管理本地数据。

* localStorage 持久化存储：*layui.data(table, settings)，数据会永久存在，除非物理删除。*
* sessionStorage 会话性存储：*layui.sessionData(table, settings)*，页面关闭后即失效。

其中参数 *table* 为表名，*settings*是一个对象，用于设置key、value。下面与 layui.data 方法为例：

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

1. //【增】：向test表插入一个nickname字段，如果该表不存在，则自动建立。
2. layui.data('test', {
3. key: 'nickname'
4. ,value: '贤心'
5. });
7. //【删】：删除test表的nickname字段
8. layui.data('test', {
9. key: 'nickname'
10. ,remove: true
11. });
12. layui.data('test', null); //删除test表
14. //【改】：同【增】，会覆盖已经存储的数据
16. //【查】：向test表读取全部的数据
17. var localTest = layui.data('test');
18. console.log(localTest.nickname); //获得“贤心”