Jlimei框架使用场景：是一个轻量级的前端开发框架。比较适用于功能不是复杂的项目。在手机上面比较适合，因为他足够小，在没有任何的压缩的情况下才30k，目前jquery的大小为273k。大小差不多在jquery的1/9. 在代码YUI压缩情况下为18k，gzip压缩有为6k

优点：体积小，扩展性强大，容易上手，如果熟悉jquery的人来用基本是不需要花什么学习成本的。目前版本也解决了ie6-9的兼容性问题。

缺点：函数库没有jquery的丰富去除很少用到的动画效果。

**操作文档：**

1.框架调用的方法列表 注解：jLimei 别名$

样本html代码：

<div id=”j\_div”>div\_html</div>

<div class=”c\_div”>c\_div\_html</div>

<div class=”c\_div”>c\_div\_html</div>

<div class=”c\_parent”><div class=”c\_child”>aaa</div></div>

动态方法的调用

方法名称：ready

可选参数（必须）：【function】

注释：当dom已经加载完成，就会发生ready事件。

语法事例：（1）$(document).ready(function (){…}); (2)$(function (){…});

方法名称：length/size(一样)

可选参数（none）：

注释：返回被Jlimei选择器匹配的元素的数量。

语法事例：$(selector).size();

代码示例：$(“.c\_div”).size();

返回结果为整型 2

方法名称：gt

可选参数（必须）：【int】

注释：获得元素index位置大于等于对象索引的集合。

语法事例：$(selector).gt(1);

代码示例：$(“.c\_div”).gt(1);

返回结果 [object object] //大于等于$(selector)的索引对象集合

方法名称：lt

可选参数（必须）：【int】

注释：获得元素index位置小于对象索引的集合。

语法事例：$(selector).lt(1);

代码示例：$(“.c\_div”).lt(1);

返回结果 [object object] //小于$(selector)的索引对象集合

方法名称：eel

可选参数（必须）：【int】

注释：获得元素index位置等于对象索引的对象。

语法事例：$(selector).eel(1);

代码示例：$(“.c\_div”).eel(1);

返回结果 [object object] //等于$(selector)的索引对象

方法名称：get

可选参数（必须）：【int】

注释：获得元素index位置等于对象索引的对象。

语法事例：$(selector).get(1);

代码示例：$(“.c\_div”).get(1);

返回结果 [object HTMLDivElement]//等于$(selector)的索引对象

方法名称：find

可选参数（必须）：【string】

注释：获得当前对象下的子对象。

语法事例：$(selector).find(selector);

代码示例：$(“.c\_parent”).find(“.c\_child”);

返回结果 [object object]

方法名称：bind

可选参数（none）：

注释：绑定一个对象。

语法事例：$(selector).bind(events,function);

代码示例：$(“#j\_div”).bind(“click”,function(){alert(“ok”);});

返回结果 Alert //ok

方法名称：unbind

可选参数（none）：

注释：移除绑定的事件。

语法事例：$(selector).bind(events,function);

代码示例：$(“#j\_div”).unbind(“click”,function(){alert(“ok”);});

返回结果 none

方法名称：width

可选参数（none）：

注释：获取当前对象的宽度。

语法事例：$(selector).width();

代码示例：var width = $(“#j\_div”).width();

返回结果 int // dom对象的宽度

方法名称：height

可选参数（none）：

注释：获取当前对象的高度。

语法事例：$(selector).height();

代码示例：var height = $(“#j\_div”).height();

返回结果 int // dom对象的宽度

方法名称：css

可选参数（key,value） / (object…[key,value])：

注释：设置当前对象的样式有单一形式，和多样设置。

语法事例：$(selector).css(key,value);

代码示例：var height = $(“#j\_div”).css(“height”,”60px”).css({“border”:”1px solid red”,width:”300px”});

返回结果 $(selector) 支持链式操作

方法名称：hide

可选参数（none）：

注释：获取当前对象的高度。

语法事例：$(selector).hide();

代码示例： var object = $(“#j\_div”).hide();

返回结果 $(selector) //隐藏了该对象

方法名称：show

可选参数（none）：

注释：获取当前对象的高度。

语法事例：$(selector).show();

代码示例：var object = $(“#j\_div”).show();

返回结果 $(selector) 显示该对象

方法名称：bodyOffset

可选参数（none）：

注释：获取跟body的偏移距离。

语法事例：$(selector). bodyOffset ();

代码示例：var object = $(“#j\_div”). bodyOffset ();

返回结果 object {top:””,left:’’} 跟body的偏移距离

方法名称：offset

可选参数（none）：

注释：获取跟文档的偏移距离。

语法事例：$(selector). offset ();

代码示例：var object = $(“#j\_div”). offset ();

返回结果 object {top:””,left:’’} 跟文档的偏移距离

方法名称：addClass

可选参数（必须）：className

注释：为元素添加className

语法事例：$(selector). addClass (“className”);

代码示例：var object = $(“#j\_div”). addClass (“className”);

返回结果 $(selector) //返回数据的原型对象

方法名称：removeClass

可选参数（必须）：className

注释：移除className

语法事例：$(selector). removeClass (“className”);

代码示例：var object = $(“#j\_div”). removeClass (“className“);

返回结果 $(selector) //返回数据的原型对象

方法名称：hasClass

可选参数（必须）：className

注释：判断参数className是否存在

语法事例：$(selector). hasClass ();

代码示例：var boolean = $(“#j\_div”). hasClass ();

返回结果 true/ false //true该对象存在，false该对象不存在

方法名称：html

可选参数（none）：

注释：获取该对象中包含html字符串

语法事例：$(selector). html ();

代码示例：var html = $(“#j\_div”). html ();

返回结果string //该对象中html文本字符串

方法名称：append

可选参数（必须）：domString/domObject

注释：获取该对象中包含html字符串，追加在当前对象里面最后位置

语法事例：$(selector). append (selector);

代码示例：var html = $(“#j\_div”). append (“<div>123</div>”);

返回结果 $(selector) //返回数据的原型对象

方法名称：prepend

可选参数（必须）：domString/domObject

注释：获取该对象中包含html字符串，追加在当前对象里面最前位置

语法事例：$(selector). prepend (selector);

代码示例：var html = $(“#j\_div”). prepend (“<div>123</div>”);

返回结果 $(selector) //返回数据的原型对象

方法名称：before

可选参数（必须）：domString/domObject

注释：获取该对象中包含html字符串，追加在当前对前面的位置

语法事例：$(selector). before (selector);

代码示例：var html = $(“#j\_div”). before (“<div>123</div>”);

返回结果 $(selector) //返回数据的原型对象

方法名称：after

可选参数（必须）：domString/domObject

注释：获取该对象中包含html字符串，追加在当前对象后面的位置

语法事例：$(selector). after (selector);

代码示例：var html = $(“#j\_div”). after (“<div>123</div>”);

返回结果 $(selector) //返回数据的原型对象

方法名称：remove

可选参数（none）

注释：移除本身当前对象

语法事例：$(selector). remove ();

代码示例： $(“#j\_div”). remove ();

返回结果 移除了#j\_div对象

方法名称：attr

可选参数（必须）：key

注释：获取该对象的属性值

语法事例：$(selector). attr (properties);

代码示例：var string = $(“#j\_div”). attr (”name”);

返回结果 string // 返回属性为name的值

方法名称：attr

可选参数（必须）：key,value

注释：设置该对象的属性值

语法事例：$(selector). attr (properties,value);

代码示例：var object = $(“#j\_div”). attr (”name”,”vincent”);

返回结果 $(selector) //返回数据的原型对象

方法名称：val

可选参数（none）：

注释：获得标签的value值

语法事例：$(selector). val ();

代码示例：var value = $(“#j\_div”). val ();

返回结果 string //或者文本框，下拉框的值

方法名称：parents

可选参数（必须）：”.tagName”

注释：获得父元素class为tagName的对象

语法事例：$(selector). parents(selector);

代码示例：var value = $(“#j\_div”). parents(“.tagName”);

返回结果 $(selector) //获得j\_div的父元素名字为tagName的对象

静态方法的调用

方法名称：Map

可选参数（none）：

注释：创建一个map对象。

语法事例： $.Map();

代码示例：var map = $.Map();

返回结果 [key:vlaue]

Map的属性详解（map是新自定义的一个对象）

Map 操作接口

var jump = $.Map();

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数 | 返回对象 | 注解 |
| jsmap.get | key | Map的对象值 |  |
| jsmap.keySet |  | Array | 返回所有的key值 |
| jsmap.values |  | Array | 返回所有的value值 |
| jsmap.size |  | Number | Hash的长度 |
| jsmap.remove | Key |  | 移除对应的key value |
| jsmap.clear |  |  | 删除所有的对象 |
| isEmpty |  |  | 清空长度为0 |
| containKey | Key | True false | 查找key是否存在 |
| toString |  | 字符串 | 将hash变成字符串 |

方法名称：StringBuffer

可选参数（none）：

注释：创建一个map对象。

语法事例：$. StringBuffer ();

代码示例：var sb = $. StringBuffer ();

返回结果 []String

StringBuffer的属性详解（sb是新自定义的一个对象）

Map 操作接口

var sb = $. StringBuffer ();

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | 参数 | 返回对象 | 注解 |
| sb. append | string | []String |  |
| sb.toString |  | String |  |

方法名称：delegate

可选参数（必须）： (selector,listenerType,fnHandler,delegates);

注释：延迟绑定 （一般用于事件未创建先绑定）

语法事例：$.delegate (selector,listenerType,fnHandler,delegates);

代码示例：$.delegate("#j\_div","click",function (\_this,event){alert(1);},”body”);

返回结果 alert // 1

方法名称：undelegate

可选参数（必须）： (selector,listenerType,fnHandler, selector);

注释：解后绑定 最后一个参数可选（如果有，就在这个对象里面的节点下面查找），没有在document下面查找。

语法事例：$.undelegate (selector,listenerType,fnHandler,delegates);

代码示例：$.undelegate("#j\_div","click",function (\_this,event){alert(1);},”body”);

返回结果 none

方法名称：addEventHandler

可选参数（必须）：([object HTMLDivElement],string,function)

注释：单个事件绑定

语法事例：$. addEventHandler([object HTMLDivElement],string,function);

代码示例：$. addEventHandler($(“#j\_div”)[0][0],”click”,function (){alert($(this).html());});

返回结果 String // div\_html

方法名称：addEventHandler

可选参数（必须）：([object HTMLDivElement],string,function)

注释：单个事件绑定

语法事例：$. removeEventHandler ([object HTMLDivElement],string,function);

代码示例：jLimei.removeEventHandler(($(“#j\_div”)[0][0],”click”,function);

返回结果 none

方法名称：getEvent

可选参数（必须）：Event

注释：单个事件绑定

语法事例：$. getEvent(event);

代码示例：var event = $. getEvent(event);

返回结果 Event //这个Event 是标准的ECMA，因为ie有部分没有根据ECMA实现。

方法名称：browerV

可选参数（none）：

注释：获取浏览器版本

语法事例：$. browerV();

代码示例：var sys = $. browerV();

返回结果 [object,object] //{ie:,””,webkit:””,firefox:””,opera:””,safari:””} 如果存在就代表为什么浏览器版本

方法名称：isFunction

可选参数（必须）：object

注释：判断是否为function

语法事例：$. isFunction(object);

代码示例：var isFun = $. isFunction(object);

返回结果 true/false

方法名称：isArray

可选参数（必须）：object

注释：判断是否为Array

语法事例：$. isArray (object);

代码示例：var isArr = $. isArray (object);

返回结果 true/false

方法名称：isWindow

可选参数（必须）：object

注释：判断是否为window

语法事例：$. isWindow(object);

代码示例：var isWin= $. isWindow(object);

返回结果 true/false

方法名称：isNumeric

可选参数（必须）：object

注释：判断是否为数字

语法事例：$. isNumeric (object);

代码示例：var isNum = $. isNumeric (object);

返回结果 true/false

方法名称：type

可选参数（必须）：object

注释：判断是否为数字

语法事例：$. type (object);

代码示例：var string = $. type (object);

返回结果 string 返回对象的类型 （string,object,array）

方法名称：parseJSON

可选参数（必须）：string

注释：解析json字符串数据

语法事例：$. parseJSON(string);

代码示例：var string = $. parseJSON(string);

返回结果 object {key,value} …

方法名称：parseXML

可选参数（必须）：string

注释：解析xml字符串数据

语法事例：$. parseXML(string);

代码示例：var string = $. parseXML(string);

返回结果 xmldoc

方法名称：globalEval

可选参数（必须）：string

注释：解析string字符串数据 用来解析字符串代码 （类似于java的反射）

语法事例：$. globalEval (string);

代码示例：var string = $. globalEval (string);

返回结果 exec code

方法名称：serializePost

可选参数（必须）：object (key,value)

注释：解析string字符串数据 用来解析字符串代码 （类似于java的反射）

语法事例：$. serializePost (object);

代码示例：var string = $. serializePost (object);

返回结果 string 类似 a=b&dd=xx&ddd=dd

方法名称：getUrlParam

可选参数（必须）： (string,string) 第二个参数可选

注释：获取url的get方式的参数值

语法事例：$. getUrlParam(string,string); 第二个参数可选

代码示例：var string = $. getUrlParam(name,domainUrl);

返回结果 string

方法名称：contains

可选参数（必须）： ([object HTMLDivElement],[ object HTMLDivElement])

注释：第二个对象是否包含在第一个对象中

语法事例：$. contains([object HTMLDivElement],[ object HTMLDivElement]);

代码示例：var bool = $. contains(oriobj,oriobj2);

返回结果 true/false

方法名称：stringToDom

可选参数（必须）： (string)

注释：把string变成一个dom对象

语法事例：$. stringToDom (string);

代码示例：var dom = $. stringToDom (string);

返回结果 [ object HTMLDivElement]

方法名称：stringify

可选参数（必须）： (object)

注释：把object变成一个可读的string

语法事例：$. stringify(object);

代码示例：var string = $. stringify(object);

返回结果 string

方法名称：checkArray

可选参数（必须）： (array,id)

注释：检查数组中是否有id这个值

语法事例：$. checkArray(Array,id);

代码示例：var string = $. checkArray(Array,id);

返回结果 true/false

方法名称：removeArrayRecord

可选参数（必须）： (array,id)

注释：移除数组中id这个值

语法事例：$. removeArrayRecord(Array,id);

代码示例：var array = $. removeArrayRecord(Array,id);

返回结果 array

方法名称：createCookie

可选参数（必须）： (name,value,days)

注释：创建cookie值 name为创建cookie的名称，value为cookie的值,days为cookie的时效

语法事例：$. createCookie (name,value,days);

代码示例：$. createCookie (name,value,days);

返回结果 none

方法名称：readCookie

可选参数（必须）： (name)

注释：获取cookie的值

语法事例：$. readCookie (name);

代码示例：var string = $. readCookie (name);

返回结果 string //cookie的值

方法名称：eraseCookie

可选参数（必须）： (name)

注释：获取cookie的值

语法事例：$. eraseCookie (name);

代码示例： $. eraseCookie (name);

返回结果 none

方法名称：clone

可选参数（必须）： (string/dom)

注释：参数为string或者dom对象，复制一个新的对象空间

语法事例：$. clone (object);

代码示例： var object = $. clone (object);

返回结果 克隆一个新的对象

方法名称：setLocalStorage

可选参数（必须）： (key,value,days)

注释：这个跟setcookie类似，这个是浏览器支持html5才支持，如果不支持，保存的方式变为cookie的保存形式

语法事例：$. setLocalStorage(key,value,days);

代码示例：$. setLocalStorage(object);

返回结果 none

方法名称：getLocalStorage

可选参数（必须）： (key)

注释：活的localStorage的值，如果浏览器不支持html5从cookie中取值

语法事例：$. getLocalStorage(key);

代码示例：var value = $. getLocalStorage(key);

返回结果 string

方法名称：clearLocalStorage

可选参数（必须）： (key)

注释：活的localStorage的值，如果浏览器不支持html5从cookie中取值

语法事例：$. clearLocalStorage(key);

代码示例：$. clearLocalStorage(“username”);

返回结果 none /清除storage为username的值

方法名称：ajax

可选参数（必须）： (object) <key,value>

注释：ajax调用支持get post

语法事例：$. ajax({…});

代码示例：$. ajax({method:”post”,isSync:false,dataType:”string”,url:’’,data:json});

返回结果 如果isSync如果为false，则返回的数据类型data的string,如果为true 需要加入一个参数callback:function

方法名称：loadScript

可选参数（必须）： (object) <key,value>

注释：loadScript，用来动态加载js文件

语法事例：$. loadScript({…});

代码示例：$. ajax({url:’’,callback:function});

返回结果none, 当url的jsload完成后调用callback:function

方法名称：jsonp

可选参数（必须）： (object) <key,value>

注释：用来做跨域调用

语法事例：$. jsonp({…});

代码示例：$. jsonp({url:’’,callback:function});

返回结果none, 当url的跨域完成时候后callback:function

方法名称：isEmail

可选参数（必须）： string

注释：用来做跨域调用

语法事例：$. isEmail(email);

代码示例：var boolean = $. isEmail();

返回结果boolean, //true 邮箱格式正确，false格式错误

基于框架扩展类

1. 对于静态方法跟动态已经在之后可能能够可能根据业务的需求对其作为扩展，现在对静态和动态api 实现接口，用来第三方作为接口的实现。

1.1 静态方法扩展接口

jLimei.extend({

fn:function(){

//code

},

Fn2:function (){

}

});

1.2 动态方法扩展接口

jLimei.fn.extend({

bind:function(types,fn){

//code

},

unbind:function(types,fn){

//code

}

});