公共全局jquery插件jquery.xhb.plugin.js

需要依赖第三方的插件库 jquery.xhb.otherPlugin.js

## Ps：插件参数类型对应表

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **数据类型** | **数组类型** | **字符串类型** | **数字类型** | **对象类型** | **函数类型** | **元素选择器** | **未定义类型** | **空类型** | **无参数类型** | **其他类型** |
| **参数指向** | Arr/arr | Str/str | N/n | Obj/obj | Fn/fn | Ele/ele | underfind | null |  | event |

# 全局插件

## A、关于判断检测

### 判断某字符串是否含有某段子字符串，有就返回其第一或者最后的索引

使用方法：$.hasStr(stringText, littleStr, isFrist)

备注：

\* 判断某字符串是否含有某个子字符串，如有，打印其第一次或者最后一次的索引

\* @param {String} stringText 指的是整个字符串变量，

\* @param {String} littleStr 指的是整个字符串变量中可能存在的字段,

\* @param {Number} isFrist 是数值，为0，则是返回第一次出现的索引，为1，则是返回最后一次的索引

\* @return {Number}

### 判断当前浏览器是否移动端

使用方法：$.isMoblie(fnMobile, fnPc)

备注：

\* 判断是否移动端，移动端执行函数1（参数1）；否则执行函数2（参数2）

\* @param {Function} fnMobile 移动端执行函数

\* @param {Function} fnPc pc端执行函数

### 检测浏览器是否支持svg

使用方法：$.isSupportSVG();

\* 检测浏览器是否支持svg

\* @return {Boolean}

### 检测浏览器是否支持canvas

使用方法：$.isSupportCanvas();

\* 检测浏览器是否支持canvas

\* @return {Boolean}

### 判断值是否为null

使用方法：$.isNull(tmp,NullFn,noNullFn);

\* 判断值是否为空，回调函数

\* @param {String} tmp 要判断的值

\* @param {Function} NullFn 为null的回调

\* @param {Function} noNullFn 为null的回调

### 判断浏览器当前版本和类型

获取浏览器类型：$.getExplorerInfo().type

获取浏览器版本：$.getExplorerInfo().version

## 关于随机生成

### 随机生成n个大写字母，数组形式返回

使用方法：$.getCapital(n)

备注： \* 随机生成n个大写字母 ,返回数组

\* @param {Number} n 字母个数

\* @return {Array}

### 随机生成范围数字

使用方法：$.randNum(min, max)

备注：

\* 随机生成范围数字：min最小数字，max最大数字（打印数字为最小到最大的范围）

\* @param {Number} min 最小值

\* @param {Number} max 最大值

\* @return {Number}

## 关于路由

### 返回路由地址

使用方法：$.urlBack(str)

### 当前页面和父页面跳转到其他页面

使用方法：$.toNewPage(Url)

备注：

\* 当前页面和父页面跳转到其他页面

\* @param {String} Url Url指的是要跳转的路劲页面，如index.html

### 父页面和当前页重新刷新加载

使用方法：$.pageReLoad()

### 返回浏览器地址上，指定的键所携带的参数值

使用方法：$.getQueryString(name);

\* 浏览器地址指定携带的参数参数，返回指定的键值

\* @param {String} name 要查询的地址参数的键

\* @return {String}

### 获取地址栏所有参数，返回json数据格式

使用方法：$.oGetParam();

\* 获取地址栏所有参数，返回json数据格式

\* @return {Object}

## 关于数据转换类型和格式

### 数组去重

使用方法：$.delRepetition(arr)

### 数组扁平化（二维数组一维处理）

使用方法：$.flattening(arr)

Ps：最短的方法实现数组去重：[...new Set([1, "1", 2, 1, 1, 3])]

### 统计数组中所有值出现的次数，并以对象形式返回

使用方法：$.countif(arr)

\* 统计数组中所有的值出现的次数,并以对象的形式返回

\* @param {Array} arr 要统计的数组

\* @return {Object}

### 字符串转数组(以字符串的某个字段截取为数组)

使用方法：$.stringToArray(str, chart)

备注：

\* 含有规律的字符串数据转化为数组；（以字符串中的某个字段截取生成数组）

\* @param {String} str 指的是大字符串

\* @param {String} chart 指的是大字符串中的某个子字符串

\* @param {Array}

### 数组排序

使用方法：$.arrSort(arr)

### yyyy-mm-dd hh:mm:ss时间格式转换为时间戳

使用方法：$.backDateNum(s);

s:指的是年月日时分秒，无时分秒，默认零时零分零秒

\* yyyy-mm-dd hh:mm:ss转换为时间戳

\* @param {String} s yyyy-mm-dd hh:mm:ss时间格式

\* @return {Number}

### 标准时间格式返回：yyyy-mm-dd hh:mm:ss格式时间

使用方法：$.formatDateTime(date);

date:指的是标准时间，泛指new Date（）出来的时间

\* 标准时间返回 y-m-d h:m:s格式

\* @param {Object} date 当前时间对象

\* @return {String}

### 转换时间戳格式返回：yyyy-mm-dd hh:mm:ss等格式时间

使用方法：$.timestampToTime(timestamp3, sFormat);

\* 时间戳转换格式

\* @param {Number} timestamp3 时间戳

\* @param {String} sFormat 要转换的数据格式

sFormat格式字符串:

'yyyy-MM-dd h:m:s'年月日时分秒

'yyyy-MM-dd'

'yyyy-MM'

'h:m'

'yyyy'

.....

### 秒数转换时分秒时间格式

使用方法：$.secondToStr(nSecond);

\* @param {Number} nSecond 秒数

\* @return {String}

### 序列化的表单数据转化为对象类型

使用方法：$.serializeToObj(param);

返回obj；

Param：序列化的form数据，通过$("form").serialize()获取的，如：name=xhb&pwd=123

\* 序列化表单的字符串转化为对象

\* @param {String} param 序列化表单的字符串

\* @return {Object}

### 数组对象中，根据对象同个键值对，分类为数组，最终返回二维数组

（数组对象，将数组中具有相同值的对象 取出组成新的数组，返回新数组）

使用方法：$.getSameVal(arr, str);

备注：

\* 数组对象，将数组中具有相同值的对象 取出组成新的数组，返回新数组

\* @param {Array} arr 数组对象

\* @param {String} str 数组对象中相同值的属性字符串

\* @return {Array}

### RSA加密（异步请求设置请求头header）(依赖RSA加密库jsrsasign-all-min.js)

使用方法：$.backHeaderParam();

使用方法如下：

$.ajax({

type: "GET",

url: "default.aspx",

headers:$.backHeaderParam(),

success: function(result) {

alert(result);

}

});

### 计算两个GPS经纬度之间的距离

使用方法：$.getDistance(lat1,lng1,lat2,lng2);

\* 获取两个GPS经纬度之间的距离

\* @param {Number} lat1 第一点的纬度

\* @param {Number} lng1 第一点的经度

\* @param {Number} lat2 第二点的纬度

\* @param {Number} lng2 第二点的经度

\* @returns {Number} 单位：米

### RSA和Crypto加密解密（异步请求设置请求头header）(依赖RSA加密解密库jsrsasign-all-min.js，crypto-js.min.js)

加密使用方法：$.backEncryptParam(); 作为请求头发送；

解密使用方法：$.backDecryptParam(data); 请求成功的回调函数使用

Ps：@param data 后台响应返回的加密数据

使用方法如下：

$.ajax({

type: "GET",

url: "default.aspx",

headers: {

http\_sign: $.backEncryptParam()

},

success: function(data) {

var result = JSON.parse($.backDecryptParam(data));

alert(result);

}

});

### 十进制转换各种进制字符（2到32进制）

使用方法：$.tenToAny(str,num);

\* 十进制转换为各种进制字符（2到32进制）

\* @param {String} str 要转换其他进制的十进制字符串

\* @param {Number} num 十进制要转换的进制数（2到32）

\* @return {String}

### 其他进制字符（2到32进制）转换为十进制字符

使用方法：$.AnyToTen(str,num);

\* 其他进制（2到32）数据转换为十进制数据

\* @param {String} str 其他进制的字符数据

\* @param {Number} num 要转换为十进制的原进制数（2到32）

\* @return {String}

## 关于获取系统相关信息

### 获取鼠标位置

使用方法：$.getMousePos(event).x；$.getMousePos(event).y

备注：

$.getMousePos(event).x==>获取鼠标水平位置

$.getMousePos(event).y==>获取鼠标垂直位置

\* 获取鼠标位置

\* @param {Event} event 事件参数标示，必传event

\* @return {Object}

### 获取当前本地的静态时间和日期

使用方法：$.getOnTime(oTime )

备注：

\* 获取当前静态所有时间

\* @param {String} oTime 时间格式参数

\* 'y-m-d' ==> 年月日

'y-m' ==> 年月

'm-d' ==> 月日

'h-m-s' ==> 时分秒

'h-m' ==> 时分

'm-s' ==> 分秒

'w' ==>星期

'' ==>年月日 时分秒

## 关于定时器优化

### 函数延时加载器

使用方法：$.debounce(fn, wait)

备注：

\* 延迟加载器

\* @param {Function} fn 回调函数

\* @param {Number} wait 时间（毫秒）

## 关于ajax请求

### 普通ajax请求

使用方法：$.ajax\_method\_1(murl, mdata, method, successFn)

\* 普通ajax请求

\* @param {String} murl 请求地址

\* @param {Object} mdata 请求数据

\* @param {String} method 请求类型

\* @param {Function} successFn 请求成功的回调函数

### 文件表单上传的ajax请求

使用方法：$.ajax\_method\_2(murl, mdata, method, successFn)

\* 文件表单上传ajax请求

\* @param {String} murl 请求地址

\* @param {Object} mdata 请求数据

\* @param {String} method 请求类型

\* @param {Function} successFn 请求成功的回调函数

### 传sessionID的ajax请求

使用方法：$.ajax\_method\_3(murl, mdata, method, successFn)

\* 传sessionID 的ajax请求

\* @param {String} murl 请求地址

\* @param {Object} mdata 请求数据

\* @param {String} method 请求类型

\* @param {Function} successFn 请求成功的回调函数

### 传sessionID的文件表单上传ajax请求

使用方法：$.ajax\_method\_4(murl, mdata, method, successFn)

\* 传sessionID 的 文件表单上传的 ajax请求

\* @param {String} murl 请求地址

\* @param {Object} mdata 请求数据

\* @param {String} method 请求类型

\* @param {Function} successFn 请求成功的回调函数

### 签名校验的ajax请求

使用方法：$.ajax\_method\_5(murl, mdata, method, successFn)

\* 加密解密的 ajax请求

\* @param {String} murl 请求地址

\* @param {Object} mdata 请求数据

\* @param {String} method 请求类型

\* @param {Function} successFn 请求成功的回调函数

### 签名校验的文件表单上传ajax请求

使用方法：$.ajax\_method\_6(murl, mdata, method, successFn)

\* 加解密的文件表单上传的 ajax请求

\* @param {String} murl 请求地址

\* @param {Object} mdata 请求数据

\* @param {String} method 请求类型

\* @param {Function} successFn 请求成功的回调函数

### 请求地址简化函数

使用方法：$.urlBack(mUrl)

\* @param {String} mUrl 后台控制器+函数

# 对象插件

## 关于表单、输入，限制匹配

### 表单特殊字符禁用验证

使用方法：$(ele).checkVerify();(默认)

备注如例：

$("input").checkVerify({

event: 'keyup', // 事件类型

paste: false, //不允许粘贴

});

### 银行表单输入格式

使用方法：$(ele).bankInput();（默认）

备注如例：

$("#account").bankInput({

min: 16, // 最少输入字数

max: 25, // 最多输入字数

deimiter: ' ', // 账号分隔符

onlyNumber: true, // 只能输入数字

copy: false, // 允许复制

paste: false, //不允许粘贴

cut: false //不允许剪切

});

### 表单中文字符禁用验证

使用方法：$(ele).checkChinese();(默认)

备注如例：

$("input").checkChinese({

event: 'keyup', // 事件类型

paste: false, //不允许粘贴

});

### 验证只能输入数字和小数

使用方法：$(ele).onlyNumAndFlo();(默认)

### 发送验证码的倒计时

使用方法：$(ele).countDown({second:100});(默认60秒)

注意：只能是input和button按钮的点击事件

### 文件上传表单上传验证（文件类型，文件大小）

使用方法：$(ele).fileVaild();(默认60秒)

如例：

$("input[type=file]").fileVaild({

event: "change", // 默认change类型

size:1,//默认最大1M

onlyImage:true,//默认只可以上传图片，false就可以上传所有类型

width:200,//默认像素至少宽度,onlyImage必须true，

height:300,//默认像素至少高度,onlyImage必须true

});

### 文本域字符输入限制

使用方法：$(ele).textareaVaild();

如例：

$("input[type=file]").fileVaild({

event:"keyup",

maxLength:300,//限制最大能输入多少字符，默认300

width:540,//文本域的宽度

height:170,//文本域 的高度

});

## 关于功能实现

### 分页插件（结合layer.js,laypage.js）

使用方法：

$(ele).paging({

url:"http://103.251.36.122/index.html" ,

data: {},

type: "post",

eleTotal:$(".nTotal"),//总数dom节点

callBack: function(data) {

Fn(data);

}

});

备注：

ele==》元素选择器

url==》服务器请求地址

data==》异步请求发送到数据（json）

type==》请求类型

eleTotal==》设置总数的元素节点选择器

callBack==》请求成功的回调（一般都是对后台数据进行渲染展示）

### 银行列表格式化

使用方法：$(ele).bankList();（默认）

备注如例：

$("li").bankList({

deimiter: ' ', // 账号分隔符

});

### 置顶

使用方法：$(ele).toTop();(默认)

备注如例：

$("button").toTop({

event: click, // 事件类型

position：[‘left’,’bottom’],

offset:[‘10px’,’10px’],

zIndex:1000,

txt:’置顶’

});

### 置底

使用方法：$(ele).toBottom();(默认)

备注如例：

$("button").toBottom({

event: click, // 事件类型

position：[‘right’,’top’],

offset:[‘10px’,’10px’],

zIndex:1000,

txt:’置底’

});

### 评级组件

使用方法：$(ele).rate();(默认)

备注如例：

$("div").rate({

star: 2, // 星星数量，最大为5，超过都为5（0--5）

edit:true //是否可编辑状态

});

### 二维码生成插件（依赖jquery.js,lib文件夹中QRcode的所有文件脚本）

使用方法：$(ele).QR\_code();(默认)

如例：

$("div").QR\_code({

event: 'click', // 事件类型

logo:'img/96.png',//二维码中间logo图片

});

### 判断鼠标滑轮方向（上下）

使用方法：$(ele).T\_or\_B();(默认)

如例：

$("#test").T\_or\_B({

toTop:function(){

console.log("向上");

},

toDown:function(){

console.log("向下");

}

});

\* 判断鼠标滑轮方向(上和下)

\* @param {Object} options 传入参数

\* @property {Function} options.toTop 鼠标向上的回调函数

\* @property {Function} options.toDown 鼠标向下的回调函数