

# NetEase Javascript Library 1.x

## API Reference (2009.11.11)

( 压缩版地址 : "http://img1.cache.netease.com/cnews/js/ntes\_jslib\_1.x.js" )

### 顶级

**\$(selector, [context])**      ( +1 重载 )

#### 函数说明

根据 CSS 选择器和上下文获取 HTML 元素。

#### 参数说明

selector

CSS 选择器。支持的选择器包括：

**\***：所有元素。

**#myId**：id 为 myId 的元素。

**E**：标签名为 E 的元素（下面简称 E 元素）。

**E.myClass**：包含 myClass 样式的 E 元素，等价于 E[class~=myClass]。

**E F**：E 的后代 F 元素。

**E > F**：E 的 F 子元素。

**E[foo]**：foo 属性值不为空的 E 元素。

**E[foo=bar]**：foo 属性值为 bar（如果属性值只有数字或包含特殊字符，请用引号括起，下同）的 E 元素。

**E[foo~=bar]**：foo 属性值中包含用空格隔开的一个或多个值且其中一个值是 bar 的 E 元素。

**E[foo!=bar]** : foo 属性值不等于 bar 的 E 元素。

context

上下文。

## 返回值

匹配到的扩展 HTML 元素 ( 数组 )。

## 代码示例一 ( 基本应用 )

```
<div id="wrapper">
  <div id="left">
    <div id="left1"></div>
    <div id="left2"></div>
  </div>
  <div id="right"></div>
</div>

<script type="text/javascript">
var divs = $("#wrapper"); // 通过 Id 选择器获取元素，切记不可遗漏“#”
divs = divs.$("&gt;div"); // $函数可链式调用。等价于"#wrapper &gt; div"
for (var i = 0, elem; elem = divs[i]; i++) { // 依次输出"left"、"right"
  alert(elem.id);
}
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="160 669 408 689" data-label="Section-Header"><h2>代码示例二 ( iframe 操作 )</h2></div><div data-bbox="195 713 903 825" data-label="Text"><pre>&lt;iframe src="blank.html" id="testFrame"&gt;&lt;/iframe&gt;

&lt;script type="text/javascript"&gt;<![CDATA[
var divsInFrame = $("div", $("#testFrame").contentWindow.document); // 获取 iframe 页
面的所有 div 元素
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="160 849 391 869" data-label="Section-Header"><h2>代码示例三 ( XML 操作 )</h2></div><div data-bbox="195 892 527 933" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[
// 读取 XML 字符串为 XML 文档对象</pre></div>
```

```

function loadXml(text) {
    var xmlDoc;
    try {
        xmlDoc = new ActiveXObject("MSXML2.DOMDocument.5.0");
        xmlDoc.loadXML(text);
    } catch(e) {
        try {
            var parser = new DOMParser();
            xmlDoc = parser.parseFromString(text, "text/xml");
        } catch (e) {
        }
    }
    if (!xmlDoc) { throw new Error("failed to create an xml document"); }
    return xmlDoc;
}

// 读取 XML
var xmlDoc = loadXml('<cities><city name="上海" code="shanghai"></city><city name="广州" code="guangzhou"></city><city name="深圳" code="shenzhen"></city></cities>');
// 获取广州节点
var guangzhou = $("city[code=guangzhou]", xmlDoc)[0];
alert(guangzhou.getAttribute("name")); // output "广州"
//]]></script>

```

**\$(elements)**      **( +1 重载 )**

## 函数说明

扩展 HTML 元素。

## 参数说明

elements

一个或多个 HTML 元素。

## 返回值

已扩展的 HTML 元素或元素数组。

## 代码示例

```
<ul>
  <li>A</li>
  <li>B</li>
</ul>
<script type="text/javascript">
var elements = $(document.getElementsByTagName("li"));    // 扩展 li 元素
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="90 385 204 408" data-label="Section-Header"><h2>多库共用</h2></div><div data-bbox="125 438 241 459" data-label="Section-Header"><h3>$.resume()</h3></div><div data-bbox="160 491 248 511" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="195 539 520 557" data-label="Text"><p>恢复本类库对$和 NTES 全局变量的占用。</p></div><div data-bbox="160 585 229 604" data-label="Section-Header"><h4>返回值</h4></div><div data-bbox="195 633 273 651" data-label="Text"><p>原$变量。</p></div><div data-bbox="125 684 221 705" data-label="Section-Header"><h3>$.retire()</h3></div><div data-bbox="160 736 248 756" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="195 784 587 802" data-label="Text"><p>恢复最近一次 NTES.resume 方法调用前的$变量。</p></div><div data-bbox="160 830 229 850" data-label="Section-Header"><h4>返回值</h4></div><div data-bbox="195 878 294 896" data-label="Text"><p>NTES 对象。</p></div>
```

# HTML 元素扩展——\$.element

## addCss(css)

### 函数说明

为当前元素添加样式。

### 参数说明

CSS

类名或内联样式字符串。

### 返回值

当前元素。

### 代码示例

```
<style type="text/css">
.pubLink { color: #ff0000; }
.pubLink:visited { color: #ff0000; }
</style>
<a href="http://www.163.com/" class="internal">网易</a>
<a href="http://www.sohu.com/" class="internal">搜狐</a>
<a href="http://www.qq.com/" class="internal">腾讯</a>
<a href="http://www.yahoo.com" class="external">雅虎</a>
<a href="http://www.amazon.com" class="external">Amazon</a>
<script type="text/javascript">
var links = $("a");
links.addCss("font-weight:bold;font-style:italic;"); // 所有链接加粗斜体
$("a.external").addCss("font-style:null;"); // 国外链接去斜体
links.addCss("pubLink"); // 所有链接都加上 pubLink 样式类
//]]&gt;&lt;/script&gt;</pre></div>
```

## removeCss(css)

### 函数说明

为当前元素删除样式。

### 参数说明

CSS

类名或内联样式字符串。

### 返回值

当前元素。

### 代码示例

```
<style type="text/css">
.pubLink { color: #ff0000; }
.pubLink:visited { color: #ff0000; }
</style>
<a href="http://www.163.com/" class="internal pubLink">网易</a>
<a href="http://www.sohu.com/" class="internal pubLink">搜狐</a>
<a href="http://www.qq.com/" class="internal pubLink">腾讯</a>
<a href="http://www.yahoo.com" class="external pubLink">雅虎</a>
<a href="http://www.amazon.com" class="external pubLink">Amazon</a>
<script type="text/javascript">
//
var links = $("a");
links.addCss("font-weight:bold;font-style:italic;");      // 所有链接加粗斜体
$("a.external").removeCss("font-style;font-weight;");    // 国外链接去粗斜体
links.removeCss("pubLink");                               // 所有链接去掉 pubLink 样式类
//]]&gt;
&lt;/script&gt;</pre></div>
```

## **addEvent(eventName, handler, data)**

### **函数说明**

为当前元素添加事件处理函数。

事件处理函数中的 this 指向元素；可使用 return false; 取消默认事件执行。

### **参数说明**

eventName

事件名。

handler

函数句柄。

data

额外传入的数据。

### **返回值**

当前元素。

### **代码示例**

```
<input type="button" value="鼠标经过弹出 1" />
<input type="button" value="鼠标经过弹出 2" />
<input type="radio" />单选 1
<input type="radio" />单选 2
<script type="text/javascript">
//
$("input[type=button]").addEvent("mouseover", function() {
    alert(this.value);
});
$("input[type=radio]").addEvent("click", function(e) {
    e.preventDefault();
    alert("click");
});
//]]&gt;</pre></div>
```

```
});  
/]]>  
</script>
```

## removeEvent(eventName, handler)

### 函数说明

移除事件处理函数。

### 参数说明

eventName

事件名。

handler

函数句柄。

### 返回值

当前元素。

### 代码示例

```
<input type="button" value="鼠标经过弹出 1" />  
<input type="button" value="鼠标经过弹出 2" />  
<script type="text/javascript">  
//<br/>var buttons = $("input[type=button]"), eventFn = function() {<br/>    alert(this.value);<br/>};<br/>buttons.addEvent("mouseover", eventFn);<br/>buttons.removeEvent("mouseover", eventFn);<br/>/]]&gt;<br/>&lt;/script&gt;</pre></div>
```



## **attr(name) ( +1 重载 )**

### **函数说明**

获取元素属性值。如果当前元素有多个，则返回第一个元素的属性值。

### **参数说明**

name

参数名。

### **返回值**

如果该属性不存在，返回 undefined，否则返回属性值。

### **代码示例**

```
<div id="test"></div>
<script type="text/javascript">
//
alert($("#test").attr("id"));    // outputs "test"
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="124 639 522 659" data-label="Section-Header"><h2><b>attr(name, value) ( +1 重载 )</b></h2></div><div data-bbox="160 691 248 711" data-label="Section-Header"><h3><b>函数说明</b></h3></div><div data-bbox="194 739 316 757" data-label="Text"><p>设置元素属性。</p></div><div data-bbox="160 784 248 804" data-label="Section-Header"><h3><b>参数说明</b></h3></div><div data-bbox="194 833 250 850" data-label="Text"><p>name</p></div><div data-bbox="228 869 299 888" data-label="Text"><p>属性名。</p></div><div data-bbox="194 906 248 924" data-label="Text"><p>value</p></div>
```

属性值或计算属性值的函数（函数中的 this 指向当前元素）。

## 返回值

当前元素。

## 代码示例

```
<div id="test">A</div>
<div>B</div>
<script type="text/javascript">
//
$("#test").attr("innerHTML", "AB");
$("#div").attr("innerHTML", function() {
    return this.innerHTML + this.innerHTML;
});
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="126 505 234 523" data-label="Section-Header"><h2>自定义扩展</h2></div><div data-bbox="160 559 833 576" data-label="Text"><p>如果要给 HTML 元素扩展其他方法，只需要把该方法动态添加到$.element 对象即可。</p></div><div data-bbox="160 603 248 623" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="195 648 751 904" data-label="Text"><pre>&lt;div id="test"&gt;abcdefg&lt;/div&gt;
&lt;script type="text/javascript"&gt;
//<![CDATA[
// 扩展一个设置元素属性的方法
NTES.element.getInnerHTML = function() {
    return NTES.util.isArray(this) ? this[0].innerHTML : this.innerHTML;
};
// 获取该 div 的 innerHTML
alert($("#test").getInnerHTML());
//]]&gt;
&lt;/script&gt;</pre></div>
```

## 工具包——\$.util

### \$.util.isArray(obj)

#### 函数说明

检查变量是否 Array 类型。

#### 参数说明

obj

指定变量。

#### 返回值

指定变量是否 Array 类型。

#### 代码示例

```
<script type="text/javascript">
//
alert(NTES.util.isArray([1, 2, 3]));      // outputs "true"
alert(NTES.util.isArray(document.getElementsByTagName("div"))); // outputs "false"
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 690 342 712" data-label="Section-Header"><h3>$.util.isFunction(obj)</h3></div><div data-bbox="160 743 248 764" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="194 791 433 810" data-label="Text"><p>检查变量是否 Function 类型。</p></div><div data-bbox="160 836 248 857" data-label="Section-Header"><h4>参数说明</h4></div><div data-bbox="194 885 231 905" data-label="Text"><p>obj</p></div><div data-bbox="228 921 316 941" data-label="Text"><p>指定变量。</p></div>
```

## 返回值

指定变量是否 Function 类型。

## 代码示例

```
<script type="text/javascript">
//
alert(NTES.util.isFunction(function() {
    alert("test");
}));    // outputs "true"
alert($.util.isFunction([1, 2, 3]));        // outputs "false"
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 410 312 431" data-label="Section-Header"><h2>$.util.toArray(col)</h2></div><div data-bbox="160 463 248 482" data-label="Section-Header"><h2>函数说明</h2></div><div data-bbox="194 511 350 528" data-label="Text"><p>把集合转换为数组。</p></div><div data-bbox="160 557 248 576" data-label="Section-Header"><h2>参数说明</h2></div><div data-bbox="194 604 228 621" data-label="Text"><p>col</p></div><div data-bbox="228 642 281 659" data-label="Text"><p>集合。</p></div><div data-bbox="160 687 229 706" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="194 735 315 753" data-label="Text"><p>转换后的数组。</p></div><div data-bbox="160 780 248 800" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="194 823 530 937" data-label="Text"><pre>&lt;a href="http://www.163.com/"&gt;网易&lt;/a&gt;
&lt;a href="http://www.sohu.com/"&gt;搜狐&lt;/a&gt;
&lt;a href="http://www.qq.com/"&gt;腾讯&lt;/a&gt;
&lt;script type="text/javascript"&gt;
//<![CDATA[</pre></div>
```

```
var links = document.getElementsByTagName("a");
alert(NTES.util.toArray(links).join(", "));
</script>
```

## **\$.util.merge(first, second)**

### **函数说明**

合并数组。此方法会影响第一个数组。

### **参数说明**

first

目标数组，必须为数组类型。

second

源数组，可以是类数组类型。

### **返回值**

目标数组。

### **代码示例**

```
<script type="text/javascript">
//
var arr1 = [1, 2, 3], arr2 = [4, 5, 6];
NTES.util.merge(arr1, arr2);
alert(arr1.join(", "));    // outputs "1,2,3,4,5,6"
//]]&gt;
&lt;/script&gt;</pre></div>
```

## **`$.util.parseTpl(tpl, [values], [isCached])`**

### **函数说明**

模板转换。

### **参数说明**

tpl

模板代码。

values

值 hashtable。

isCached

是否缓存模板，默认为缓存。

### **返回值**

转换后的代码。

### **代码示例**

```
<div id="test"><!--
<ul>
<# for (var i = 0; i < datas.length; i++) { #>
    <li><a href="http://<#=datas[i].url#>"><#=datas[i].title#></a></li>
<# } #>
</ul>
--></div>
<script type="text/javascript">
//<![CDATA[
var values = {
    datas : [
        { url : "g.cn", title : "谷歌" },
        { url : "163.com", title: "网易" },
```

```
        { url : "discuz.com", title : "康盛" }
    ]
};
$("#test").innerHTML = NTES.util.parseTpl(
    $("#test").innerHTML.replace(/^\s*<!--([\W\w]*?)-->\s*$/, "$1"),    // 把注释符干掉
    values
);
//]]>
</script>
```

## **\$.util.extend(des, src)**

### **函数说明**

把源对象的属性和方法扩展到目标对象上。

### **参数说明**

des

目标对象。

src

源对象。

### **返回值**

已扩展的目标对象。

### **代码示例**

```
<script type="text/javascript">
//
var objA = { a : 1,b : 2 }, objB = { c : 3, d : 4};
objA = NTES.util.extend(objA, objB);
alert(objA.c);    // ouputs "3"
//]]&gt;
&lt;/script&gt;</pre></div>
```

# DOM——\$.dom

## \$.dom.eachNode(nodes, callback, args)

### 函数说明

对一个或多个节点执行某个函数。

### 参数说明

nodes

节点或节点数组、节点集合。

callback

处理函数（函数中的 this 指向节点本身）。

args

函数参数。

### 返回值

当前节点。

### 代码示例

```
<div>1</div>
<div>2</div>
<script type="text/javascript">
//
// 把所有 div 的 innerHTML 设为 test
NTES.dom.eachNode(document.getElementsByTagName("div"), function() {
    this.innerHTML = "test";
});
//]]&gt;
&lt;/script&gt;</pre></div>
```



## 样式控制——\$.style

### \$.style.addCss(elems, css)

#### 函数说明

为指定 HTML 元素添加样式 ( element.addCss 的另一种调用方式 )。

#### 参数说明

elems

一个或多个 HTML 元素。

css

CSS 类名或样式字符串。

#### 返回值

当前元素。

#### 代码示例

```
<div id="test"></div>
<script type="text/javascript">
//
NTES.style.addCss($("#test"), "background:#ff0000;width:30px;height:30px;");
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="126 765 431 786" data-label="Section-Header"><h3>$.style.removeCss(elems, css)</h3></div><div data-bbox="161 817 248 837" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="195 865 764 884" data-label="Text"><p>为指定 HTML 元素删除样式 ( element.removeCss 的另一种调用方式 )。</p></div>
```

## 参数说明

elems

一个或多个 HTML 元素。

CSS

CSS 类名或样式字符串。

## 返回值

当前元素。

## 代码示例

```
<div id="test"></div>
<script type="text/javascript">
//
NTES.style.removeCss($("#test"), "background;width;height;");
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 565 561 587" data-label="Section-Header"><h2>$.style.getCurrentStyle(nodes, styleName)</h2></div><div data-bbox="160 618 248 638" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 667 404 685" data-label="Text"><p>获取指定节点的当前样式。</p></div><div data-bbox="160 712 248 732" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 762 254 778" data-label="Text"><p>nodes</p></div><div data-bbox="229 798 740 816" data-label="Text"><p>一个或多个节点。如果有多个节点，则只取第一个节点的样式值。</p></div><div data-bbox="195 835 294 853" data-label="Text"><p>styleName</p></div><div data-bbox="229 872 299 890" data-label="Text"><p>样式名。</p></div>
```

## 返回值

样式值。

## 代码示例

```
<div id="test"></div>
<script type="text/javascript">
//
alert(NTES.style.getCurrentStyle($("#test"), "display"));    // outputs "block"
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="89 360 357 384" data-label="Section-Header"><h2>事件控制——$.event</h2></div><div data-bbox="125 414 266 435" data-label="Section-Header"><h3>$.event.fix(e)</h3></div><div data-bbox="160 467 248 486" data-label="Section-Header"><h2>函数说明</h2></div><div data-bbox="194 516 333 533" data-label="Text"><p>处理事件兼容性。</p></div><div data-bbox="160 561 248 580" data-label="Section-Header"><h2>参数说明</h2></div><div data-bbox="194 610 214 626" data-label="Text"><p>e</p></div><div data-bbox="228 645 315 663" data-label="Text"><p>事件对象。</p></div><div data-bbox="160 691 229 711" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="194 739 421 757" data-label="Text"><p>经过兼容性处理的事件对象。</p></div><div data-bbox="160 784 248 804" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="194 828 549 942" data-label="Text"><pre>&lt;input type="button" value="test" id="test" /&gt;
&lt;script type="text/javascript"&gt;
//<![CDATA[
$("#test").onclick = function(e) {
    e = NTES.event.fix(e || window.event);</pre></div>
```

```
    alert(e.pageX);  
};  
//]]>  
</script>
```

## **\$.event.addEvent(elems, eventName, handler, data)**

### **函数说明**

为指定 HTML 元素添加事件处理函数（element.addEvent 的另一种调用方式）。

### **参数说明**

elems

HTML 元素、HTML 元素集合或 HTML 元素数组。

eventName

事件名。

handler

函数句柄。

data

额外传入的数据。

### **返回值**

指定元素。

### **代码示例**

```
<input type="button" id="test" value="test" />  
<script type="text/javascript">  
//<br/>NTES.event.addEvent($("#test"), "click", function() {<br/>    alert(this.value);<br/>});</pre></div>
```

```
//]]>  
</script>
```

## **\$.event.removeEvent(elems, eventName, handler)**

### **函数说明**

为指定 HTML 元素移除事件处理函数 ( element.removeEvent 的另一种调用方式 )。

### **参数说明**

elems

一个或多个 HTML 元素。

eventName

事件名。

handler

函数句柄。

### **返回值**

指定元素。

### **代码示例**

```
<input type="button" id="test" value="test" />  
<script type="text/javascript">  
  //<br/>    var eventFn = function() { alert(this.value); };<br/>    NTES.event.addEvent($("#test"), "click", eventFn);<br/>    NTES.event.removeEvent($("#test"), "click", eventFn);<br/>  //]]&gt;<br/>&lt;/script&gt;</pre></div>
```

# 浏览器判断——\$.browser

## \$.browser

### 对象说明

获取当前浏览器的类型以及版本。

### 属性说明

version

浏览器版本号。

safari

是否 safari 浏览器。

opera

是否 opera 浏览器。

msie

是否 IE 浏览器。

mozilla

是否 Mozilla 浏览器。

### 代码示例

```
<script type="text/javascript">
//
if (NTES.browser.msie) {
    alert(NTES.browser.version);
}
//]]&gt;
&lt;/script&gt;</pre></div>
```

## 动态加载——\$.ajax

### \$.ajax.createXhr()

#### 函数说明

创建 XMLHttpRequest 对象。

#### 返回值

XMLHttpRequest 对象。

#### 代码示例

```
<script type="text/javascript">
//
var xhr = NTES.ajax.createXhr();
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 536 634 557" data-label="Section-Header"><h3>$.ajax.send(url, [method], [data], [options], [xhr])</h3></div><div data-bbox="161 589 248 608" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="195 637 325 656" data-label="Text"><p>发送 Ajax 请求。</p></div><div data-bbox="160 684 248 702" data-label="Section-Header"><h4>参数说明</h4></div><div data-bbox="195 731 225 748" data-label="Text"><p>url</p></div><div data-bbox="230 768 315 785" data-label="Text"><p>请求地址。</p></div><div data-bbox="195 805 268 821" data-label="Text"><p>method</p></div><div data-bbox="230 842 438 860" data-label="Text"><p>发送方式，POST 或 GET。</p></div><div data-bbox="195 879 240 896" data-label="Text"><p>data</p></div><div data-bbox="230 915 333 933" data-label="Text"><p>发送的数据。</p></div>
```

options

其他可选参数。常用参数包括：

headers：请求头；

timeout：超时时间；

onSuccess：请求成功后的操作；

onError：请求失败时的操作；

onTimeout：请求超时时的操作。

xhr

用于发送请求的 XMLHttpRequest 对象，如不指定则自动创建。

## 返回值

进行本次 Ajax 请求的 XMLHttpRequest 对象。

## 代码示例

```
<script type="text/javascript">
//
NTES.ajax.send(
    "ajaxtest.asp",
    "POST",
    { username : "tom", password : "iamtom" },
    {
        headers : { "Content-Type" : "text/html;charset=GBK" },    // 指定为 GBK 编码
        timeout : 30000,      // 超时时间 30 秒
        onSuccess : function(xhr) {
            alert(xhr.responseText);    // 请求成功后输出返回的文本
        },
        onTimeout : function() {
            alert("timeout");
        },
        onError : function(xhr) {
            alert("error");
        }
    }
);
//]]&gt;</pre></div>
```



```
    }  
  }  
);  
//]]>  
</script>
```

## **\$.ajax.importJs(url, [onComplete], [charset])**

### **函数说明**

加载一个外部 Javascript 脚本文件。

### **参数说明**

url

请求地址。

onComplete

加载完成后执行的函数。

charset

文件编码。

### **代码示例**

```
<script type="text/javascript">  
//<br/>NTES.ajax.importJs(<br/>    "test.js",<br/>    function() {<br/>        alert("loaded");<br/>    },<br/>    "gb2312"<br/>);<br/>//]]&gt;<br/>&lt;/script&gt;</pre></div>
```

# Cookie 操作——\$.cookie

## \$.cookie.encoder

### 函数说明

cookie 编码函数 ( 可设置项 ), 默认为 encodeURIComponent。

## \$.cookie.decoder

### 函数说明

cookie 解码函数 ( 可设置项 ), 默认为 decodeURIComponent。

## \$.cookie.get(name, allowUndefined)

### 函数说明

获取 Cookie 值。

### 参数说明

name

Cookie 名。

allowUndefined

指定该 cookie 不存在时是返回 undefined 还是空字符串 ( 参数值为 true , 则返回 undefined , 否则返回空字符串 )。

### 返回值

Cookie 值。

## 代码示例

```
<script type="text/javascript">
//
alert(NTES.cookie.get("userName"));
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 245 762 266" data-label="Text"><p><b><code>$.cookie.set(name, value, [expires], [domain], [path], [secure])</code></b></p></div><div data-bbox="160 297 248 317" data-label="Section-Header"><h2>函数说明</h2></div><div data-bbox="194 346 330 363" data-label="Text"><p>设置 Cookie 值。</p></div><div data-bbox="160 391 248 410" data-label="Section-Header"><h2>参数说明</h2></div><div data-bbox="194 439 250 455" data-label="Text"><p>name</p></div><div data-bbox="229 476 326 493" data-label="Text"><p>Cookie 名。</p></div><div data-bbox="194 513 248 529" data-label="Text"><p>value</p></div><div data-bbox="229 550 326 567" data-label="Text"><p>Cookie 值。</p></div><div data-bbox="194 587 255 605" data-label="Text"><p>expire</p></div><div data-bbox="229 623 915 642" data-label="Text"><p>过期时间，默认单位为分钟。可指定单位：y - 年，365 天；M - 月，30 天；d - 日；h</p></div><div data-bbox="200 660 411 678" data-label="Text"><p>- 小时；m - 分钟；s - 秒。</p></div><div data-bbox="194 697 268 715" data-label="Text"><p>domain</p></div><div data-bbox="229 734 263 752" data-label="Text"><p>域。</p></div><div data-bbox="194 771 242 790" data-label="Text"><p>path</p></div><div data-bbox="229 808 281 826" data-label="Text"><p>路径。</p></div><div data-bbox="194 846 258 863" data-label="Text"><p>secure</p></div><div data-bbox="229 882 629 901" data-label="Text"><p>是否仅把 Cookie 发送给受保护的服务器（https）。</p></div>
```

## 代码示例

```
<script type="text/javascript">
//
NTES.cookie.set("userName", "heero", "3M");    // 保存 3 个月
NTES.cookie.set("password", "123456", 5);      // 保存 5 分钟
alert(NTES.cookie.get("userName"));    // output "heero"
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 293 504 313" data-label="Section-Header"><h2><b>$.cookie.del(name, [domain], [path])</b></h2></div><div data-bbox="160 345 248 365" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="194 394 330 411" data-label="Text"><p>删除 Cookie 值。</p></div><div data-bbox="160 438 248 458" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="194 488 250 503" data-label="Text"><p>name</p></div><div data-bbox="229 524 326 541" data-label="Text"><p>Cookie 名。</p></div><div data-bbox="194 561 267 578" data-label="Text"><p>domain</p></div><div data-bbox="229 598 263 616" data-label="Text"><p>域。</p></div><div data-bbox="194 635 241 653" data-label="Text"><p>path</p></div><div data-bbox="229 672 280 689" data-label="Text"><p>路径。</p></div><div data-bbox="160 717 248 737" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="194 761 601 922" data-label="Text"><pre>&lt;script type="text/javascript"&gt;
//<![CDATA[
NTES.cookie.set("userName", "heero");
NTES.cookie.del("userName");
alert(NTES.cookie.get("userName"));    // output ""
//]]&gt;
&lt;/script&gt;</pre></div>
```

## 界面交互——\$.ui

**\$.ui.Slide([ctrls], contents, css, [eventName], [interval])**

### 类说明

切换控制类。

### 构造函数参数说明

ctrls

控制切换的元素数组。

contents

内容元素数组。

css

切换时附加的 CSS。

eventName

触发切换的事件名。

interval

播放间隔，0 为不播放。

### 属性

**playMode(curIndex, total)**

播放模式（产生下一帧序号的函数，传入的两个参数分别是当前帧序号和总帧数）。

**rollbackMode(curIndex, total)**

回滚模式（产生上一帧序号的函数，传入的两个参数分别是当前帧序号和总帧数）。

## **onShow(index, ctrl, content)**

切换时触发的事件( 传入的三个参数分别是当前帧序号 ,当前帧控制项 ,当前帧内容项 )。

## **方法**

### **show(seq)**

函数说明

显示某一项。

参数说明

seq

序号。

### **showNext()**

显示下一项。

### **showPrevious()**

显示上一项。

### **play(playInterval)**

函数说明

开始自动播放。

参数说明

playInterval

播放间隔。

### **pause()**

暂停自动播放。

## 代码示例一（简单播放）

```
<style type="text/css">
#ctrls li.current { font-weight: bold; }
#contents div { display: none; }
#contents div.current { display: block; }
</style>
<ul id="ctrls">
<li class="current">1</li>
<li>2</li>
<li>3</li>
<li>4</li>
</ul>
<div id="contents">
<div class="current">A</div>
<div>B</div>
<div>C</div>
<div>D</div>
</div>
<script type="text/javascript">
//
new NTES.ui.Slide($("#ctrls &gt; li"), $("#contents &gt; div"), "current", "mouseover", 3000); //
以每 3 秒切换一次的速度自动播放
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="160 671 384 690" data-label="Section-Header"><h2>代码示例二（联动播放）</h2></div><div data-bbox="194 715 694 946" data-label="Text"><pre>&lt;style type="text/css"&gt;
&lt;!--
#ctrls li.current { color: #ff0000; }
#contents1 div, #contents2 div { display: none; }
#contents1 div.current, #contents2 div.current { display: block; }
--&gt;
&lt;/style&gt;
&lt;ul id="ctrls"&gt;
    &lt;li class="current"&gt;1&lt;/li&gt;
    &lt;li&gt;2&lt;/li&gt;</pre></div>
```

```

        <li>3</li>
        <li>4</li>
    </ul>
    <div id="contents1">
        <div class="current">A1</div>
        <div>B1</div>
        <div>C1</div>
        <div>D1</div>
    </div>
    <div id="contents2">
        <div class="current">A1</div>
        <div>B1</div>
        <div>C1</div>
        <div>D1</div>
    </div>
    <script type="text/javascript">
    //
    var slide1 = new NTES.ui.Slide($("#ctrls&gt;li"), $("#contents1&gt;div"), "current", "mouseover");
    var slide2 = new NTES.ui.Slide(null, $("#contents2&gt;div"), "current");
    slide1.onShow = slide2.show.bind(slide2);    // 同时显示 slide2 中的相应项，达到同步效果
    slide1.play(1000);    // 开始播放
    //]]&gt;
    &lt;/script&gt;
</pre>
</div>
<div data-bbox="160 623 484 643" data-label="Section-Header">
<h3>代码示例三（点击图片跳到下一张）</h3>
</div>
<div data-bbox="194 668 649 948" data-label="Text">
<pre>
&lt;style type="text/css"&gt;
#pics li { display: none; cursor : pointer; }
#pics li.current { display: block; }
&lt;/style&gt;
&lt;ul id="pics"&gt;
    &lt;li class="current"&gt;&lt;img src="1.jpg" alt="pic1" /&gt;&lt;/li&gt;
    &lt;li&gt;&lt;img src="2.jpg" alt="pic2" /&gt;&lt;/li&gt;
    &lt;li&gt;&lt;img src="3.jpg" alt="pic3" /&gt;&lt;/li&gt;
    &lt;li&gt;&lt;img src="4.jpg" alt="pic4" /&gt;&lt;/li&gt;
&lt;/ul&gt;
&lt;script type="text/javascript"&gt;
//<![CDATA[
</pre>
</div>
```



```

var items = $("#pics > li"), imgs = items.$("img");
var picSlide = new NTES.ui.Slide(null, items, "current");
picSlide.show(0); // 显示第一张
imgs.addEvent("click", picSlide.showNext.bind(picSlide));
//]]>
</script>

```

#### 代码示例四（用 select 控件控制显示隐藏）

```

<style type="text/css">
#contents li {
    display: none;
}
#contents li.current {
    display: block;
}
</style>
<select id="menu">
    <option>一</option>
    <option>二</option>
    <option>三</option>
</select>
<ul id="contents">
    <li class="current">内容 1</li>
    <li>内容 2</li>
    <li>内容 3</li>
</ul>
<script type="text/javascript">
//
var slide = new NTES.ui.Slide(null, $("#contents &gt; li"), "current"), menu = $("#menu");
menu.addEvent(
    "change",
    function() {
        slide.show(this.selectedIndex);
    }
);
//]]&gt;
&lt;/script&gt;
</pre>
</div>
```

# String 类扩展

## String.prototype.trim([str])

### 函数说明

去掉当前字符串两端的某段字符串，默认为去掉两端的空白。

### 参数说明

str

要去掉的字符串。

### 返回值

修整后的字符串。

### 代码示例

```
<input type="text" id="userName" />
<input type="button" value="Check" />
<script type="text/javascript">
//
$("input[type=button]").addEvent("click", function() {
    if (" " == $("#userName").value.trim()) { // 去掉字符串前后的空白后再判断
        alert("请输入用户名");
    }
});
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="125 810 515 832" data-label="Section-Header"><h2>String.format(str, arg1, arg2, ..., argN)</h2></div><div data-bbox="161 863 248 883" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 910 315 929" data-label="Text"><p>格式化字符串。</p></div>
```

## 参数说明

str

要格式化的字符串。

argN (N > 0)

参数值

## 返回值

格式化后的字符串。

## 代码示例

```
<script type="text/javascript">
//
alert(String.format("%1, %2, %3", "a", "b", "c")); // output "a, b, c"
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="89 541 255 566" data-label="Section-Header"><h1>Array 类扩展</h1></div><div data-bbox="125 595 556 616" data-label="Section-Header"><h2>Array.prototype.indexOf(elem, [startPos])</h2></div><div data-bbox="160 647 248 667" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 695 421 713" data-label="Text"><p>在当前数组中检索指定元素。</p></div><div data-bbox="160 741 248 761" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 790 244 807" data-label="Text"><p>elem</p></div><div data-bbox="229 826 315 844" data-label="Text"><p>指定元素。</p></div><div data-bbox="195 863 271 880" data-label="Text"><p>startPos</p></div><div data-bbox="229 899 507 918" data-label="Text"><p>开始检索的位置，0 是第一个位置。</p></div>
```

## 返回值

指定元素在数组中第一个匹配项的索引；如果该元素不存在于数组中，返回-1。

## 代码示例

```
<script type="text/javascript">
//
var arr = [1, 2, 3, 4, 1];
alert(arr.indexOf(1, 2)); // output 4
//]]&gt;
&lt;/script&gt;</pre></div><div data-bbox="90 360 298 385" data-label="Section-Header"><h2>Function 类扩展</h2></div><div data-bbox="125 414 635 436" data-label="Section-Header"><h3>Function.prototype.bind(obj, arg1, arg2, ..., argN)</h3></div><div data-bbox="160 467 248 486" data-label="Section-Header"><h2>函数说明</h2></div><div data-bbox="194 515 389 532" data-label="Text"><p>为函数绑定 this 和参数。</p></div><div data-bbox="160 560 248 580" data-label="Section-Header"><h2>参数说明</h2></div><div data-bbox="194 608 231 628" data-label="Text"><p>obj</p></div><div data-bbox="228 645 390 663" data-label="Text"><p>指定为 this 的对象。</p></div><div data-bbox="194 683 310 701" data-label="Text"><p>argN (N &gt; 0)</p></div><div data-bbox="228 720 315 737" data-label="Text"><p>函数参数。</p></div><div data-bbox="160 765 229 785" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="194 812 408 831" data-label="Text"><p>绑定了 this 和参数的函数。</p></div><div data-bbox="160 857 248 878" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="194 901 434 944" data-label="Text"><pre>&lt;script type="text/javascript"&gt;
//<![CDATA[</pre></div>
```

```
function test(arg) {  
    alert(this.value + arg);  
}  
var obj = {  
    value : 1  
};  
test.bind(obj, 2)();    // output "3"  
/]]>  
</script>
```