

jRaiser Javascript Library

API Reference (2009-11-28)

核心

\$(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

{String} 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 + F: 紧随 E 元素之后的同级元素 F。

E ~ F: 任何在 E 元素之后的同级 F 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。

E[foo^=bar]: foo 属性值以 bar 开头的 E 元素。

E[foo\$=bar]: foo 属性值以 bar 结尾的 E 元素。

E[foo*=bar]: foo 属性值包含 bar 的 E 元素。

E:first-child: E 元素的第一个子元素。

E:nth-child(n): E 元素的第 n 个子元素。

E:last-child: E 元素的最后一个子元素。

E:nth-last-child(n): E 元素的倒数第 n 个子元素。

E:odd: 第偶数个 E 元素。

E:even: 第奇数个 E 元素。

E:lt(n): 第 n 个 E 元素前的 E 元素。

E:gt(n): 第 n 个 E 元素后的 E 元素。

context

{[HtmlElement](#),[HtmlElementArray](#),[HtmlCollection](#)} 上下文。

返回值

匹配到的经扩展的 HTML 元素。

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="160 761 405 779" data-label="Section-Header"><h2>代码示例二（iframe 操作）</h2></div><div data-bbox="195 793 903 906" 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>
```

\$(elements) (+1 重载)

函数说明

扩展 HTML 元素。

参数说明

elements

`{HTMLElement, HTMLElementArray, HtmlCollection}` 要扩展的 Html 元素。

返回值

`{HTMLElement, HTMLElementArray}` 经扩展的 HTML 元素。

代码示例

```
<ul>
  <li>A</li>
  <li>B</li>
</ul>
<script type="text/javascript">
var elements = $(document.getElementsByTagName("li"));    // 扩展 li 元素
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="92 532 218 557" data-label="Section-Header"><h2>类库信息</h2></div><div data-bbox="125 580 237 600" data-label="Section-Header"><h3><b>$.version</b></h3></div><div data-bbox="161 621 248 640" data-label="Section-Header"><h4>属性说明</h4></div><div data-bbox="196 655 384 672" data-label="Text"><p><code>{String}</code> 标识当前版本。</p></div><div data-bbox="92 692 218 717" data-label="Section-Header"><h2>多库共用</h2></div><div data-bbox="125 740 252 762" data-label="Section-Header"><h3><b>$.resume()</b></h3></div><div data-bbox="161 782 248 801" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="196 814 531 832" data-label="Text"><p>恢复本类库对$和JRaiser全局变量的占用。</p></div><div data-bbox="161 846 229 865" data-label="Section-Header"><h4>返回值</h4></div><div data-bbox="196 879 343 896" data-label="Text"><p><code>{Mixed}</code> 原$变量。</p></div>
```

\$.retire()

函数说明

恢复最近一次本类库加载前或 jRaiser.resume 方法调用前的\$变量。

返回值

{Object} jRaiser 对象。

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

{String} 类名或嵌入样式字符串，多个类名用空格隔开。

返回值

{HtmlElement, HtmlElementArray} 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

`{String}` 类名或内联样式，多个类名用空格隔开。

返回值

`{HTMLElement, HTMLElementArray}` 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><div data-bbox="125 727 555 748" data-label="Section-Header"><h2>addEvent(eventName, handler[, data])</h2></div><div data-bbox="161 769 248 787" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 802 440 818" data-label="Text"><p>为当前元素添加事件委托函数。</p></div><div data-bbox="196 832 800 849" data-label="Text"><p>事件委托函数中的 <code>this</code> 指向元素；可使用 “<code>return false;</code>” 取消默认事件执行。</p></div><div data-bbox="161 862 248 880" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 896 302 911" data-label="Text"><p>eventNames</p></div><div data-bbox="230 925 544 943" data-label="Text"><p><code>{String}</code> 事件名，多个事件用逗号隔开。</p></div>
```

handler

{Function} 事件委托函数。

data

{Mixed} 额外传入的数据。

返回值

{HTMLElement, HTMLElementArray} 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;
&lt;/script&gt;</pre></div><div data-bbox="126 598 521 620" data-label="Section-Header"><h2>removeEvent(eventName, handler)</h2></div><div data-bbox="160 640 248 659" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 673 352 689" data-label="Text"><p>移除事件委托函数。</p></div><div data-bbox="160 703 248 722" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 737 291 753" data-label="Text"><p>eventName</p></div><div data-bbox="230 766 542 784" data-label="Text"><p><b>{String}</b> 事件名，多个事件用逗号隔开。</p></div><div data-bbox="196 796 263 812" data-label="Text"><p>handler</p></div><div data-bbox="230 824 441 842" data-label="Text"><p><b>{Function}</b> 事件委托函数。</p></div><div data-bbox="160 855 229 875" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="196 888 541 906" data-label="Text"><p><b>{HTMLElement, HTMLElementArray}</b> 当前元素。</p></div>
```

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">
var buttons = $("input[type=button]"), eventFn = function() {
    alert(this.value);
};
buttons.addEvent("mouseover", eventFn);
buttons.removeEvent("mouseover", eventFn);
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 317 428 339" data-label="Section-Header"><h2>attr(name) (+1 重载)</h2></div><div data-bbox="161 360 248 379" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 392 794 409" data-label="Text"><p>获取当前元素的属性值。如果当前元素有多个，则返回第一个元素的属性值。</p></div><div data-bbox="161 423 248 442" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 457 247 472" data-label="Text"><p>name</p></div><div data-bbox="229 486 366 503" data-label="Text"><p>{String} 属性名。</p></div><div data-bbox="161 517 229 535" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="195 550 332 567" data-label="Text"><p>{Mixed} 属性值。</p></div><div data-bbox="161 579 248 599" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="195 612 534 702" data-label="Text"><pre>&lt;div id="test"&gt;&lt;/div&gt;
&lt;script type="text/javascript"&gt;<![CDATA[
alert($("#test").attr("id")); // outputs "test"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 723 498 744" data-label="Section-Header"><h2>attr(name, value) (+1 重载)</h2></div><div data-bbox="161 765 248 783" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 797 387 814" data-label="Text"><p>设置当前元素的属性值。</p></div><div data-bbox="161 828 248 847" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 862 247 878" data-label="Text"><p>name</p></div><div data-bbox="229 891 366 909" data-label="Text"><p>{String} 属性名。</p></div><div data-bbox="195 920 246 937" data-label="Text"><p>value</p></div>
```

`{String,Function}` 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

`{HTMLElement,HTMLElementArray}` 当前元素。

代码示例

```
<div id="test">A</div>
<div>B</div>
<script type="text/javascript">
$("#test").attr("innerHTML", "AB");
$("#div").attr("innerHTML", function() {
    return this.innerHTML + this.innerHTML;
});
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="128 389 252 409" data-label="Section-Header"><h2>自定义扩展</h2></div><div data-bbox="161 432 827 449" data-label="Text"><p>如果要给 HTML 元素扩展其他方法，只需要把该方法动态添加到 <code>$.element</code> 对象即可。</p></div><div data-bbox="161 463 248 482" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="195 496 714 704" 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
$.element.getInnerHTML = function() {
    return $.util.isArray(this) ? this[0].innerHTML : this.innerHTML;
};
// 获取该 div 的 innerHTML
alert($("#test").getInnerHTML());
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 723 321 748" data-label="Section-Header"><h1>工具包——$.util</h1></div><div data-bbox="125 770 352 793" data-label="Section-Header"><h2>$.util.isArray(value)</h2></div><div data-bbox="161 812 248 832" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 845 403 864" data-label="Text"><p>检查变量是否 <code>Array</code> 类型。</p></div><div data-bbox="161 875 248 895" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 909 246 926" data-label="Text"><p>value</p></div>
```


{Mixed} 待测变量。

返回值

{Boolean} 待测变量是否 **Array** 类型。

代码示例

```
<script type="text/javascript">
alert($.util.isArray([1, 2, 3]));      // outputs "true"
alert($.util.isArray(document.getElementsByTagName("div"))); // outputs "false"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 293 389 314" data-label="Section-Header"><h2><b>$.util.isFunction(value)</b></h2></div><div data-bbox="160 335 248 354" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 367 387 384" data-label="Text"><p>检查变量是否函数类型。</p></div><div data-bbox="160 397 248 416" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 432 227 449" data-label="Text"><p>obj</p></div><div data-bbox="230 461 384 478" data-label="Text"><p><b>{Mixed}</b> 待测变量。</p></div><div data-bbox="160 491 229 510" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="195 525 472 542" data-label="Text"><p><b>{Boolean}</b> 待测变量是否函数类型。</p></div><div data-bbox="160 555 248 574" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="195 588 647 725" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[
alert($.util.isFunction(function() {
    alert("test");
}));      // outputs "true"
alert($.util.isFunction([1, 2, 3]));           // outputs "false"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 746 328 767" data-label="Section-Header"><h2><b>$.util.toArray(col)</b></h2></div><div data-bbox="160 787 248 806" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 820 352 838" data-label="Text"><p>把集合转换为数组。</p></div><div data-bbox="160 851 248 870" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 885 227 901" data-label="Text"><p>col</p></div><div data-bbox="230 913 425 932" data-label="Text"><p><b>{Array,Collection}</b> 集合。</p></div>
```

返回值

{Array} 转换后的数组。

代码示例

```
<a href="http://www.163.com/">网易</a>
<a href="http://www.sohu.com/">搜狐</a>
<a href="http://www.qq.com/">腾讯</a>
<script type="text/javascript">
var links = document.getElementsByTagName("a");
alert($.util.toArray(links).join(", "));
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="126 334 421 355" data-label="Section-Header"><h2><b>$.util.merge(first, second)</b></h2></div><div data-bbox="161 376 248 394" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 408 282 424" data-label="Text"><p>合并数组。</p></div><div data-bbox="161 439 248 458" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 473 234 488" data-label="Text"><p>first</p></div><div data-bbox="230 502 521 519" data-label="Text"><p><b>{Array}</b> 目标数组，必须为数组类型。</p></div><div data-bbox="195 532 260 547" data-label="Text"><p>second</p></div><div data-bbox="230 561 602 579" data-label="Text"><p><b>{Array,Collection}</b> 源数组，可以是数组或集合。</p></div><div data-bbox="161 592 229 610" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="195 625 415 642" data-label="Text"><p><b>{Array}</b> 混合后的目标数组。</p></div><div data-bbox="161 655 248 673" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="195 688 552 801" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[
var arr1 = [1, 2, 3], arr2 = [4, 5, 6];
$.util.merge(arr1, arr2);
alert(arr1.join(", "));    // outputs "1,2,3,4,5,6"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="126 821 557 843" data-label="Section-Header"><h2><b>$.util.parseTpl(tpl, values[, isCached])</b></h2></div><div data-bbox="161 863 248 882" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 895 282 912" data-label="Text"><p>模板转换。</p></div>
```

参数说明

tpl

`{String}` 模板代码。

values

`{Object}` 值集合。

isCached

`{Boolean}` 是否缓存模板，默认为缓存。

返回值

`{String}` 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">
var values = {
    datas : [
        { url : "g.cn", title : "谷歌" },
        { url : "163.com", title : "网易" },
        { url : "discuz.com", title : "康盛" }
    ]
};

$("#test").innerHTML = $.util.parseTpl(
    $("#test").innerHTML.replace(/(?:\s*!-)|(?:--\s*)/g, ""),    // 把注释符干掉
    values
);
//]]&gt;&lt;/script&gt;</pre></div>
```

\$.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

{Mixed} 目标对象。

src

{Mixed} 源对象。

返回值

{Mixed} 已扩展的目标对象。

代码示例

```
<script type="text/javascript"><br/>var objA = { a : 1,b : 2 }, objB = { c : 3, d : 4};<br/>objA = $.util.extend(objA, objB);<br/>alert(objA.c); // ouputs "3"<br/>//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 544 318 567" data-label="Section-Header"><h2><b>DOM——$.dom</b></h2></div><div data-bbox="125 592 591 613" data-label="Section-Header"><h3><b>$.dom.eachNode(nodes, callback[, args])</b></h3></div><div data-bbox="161 634 248 652" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="196 667 405 683" data-label="Text"><p>对指定节点执行某个函数。</p></div><div data-bbox="161 697 248 715" data-label="Section-Header"><h4>参数说明</h4></div><div data-bbox="196 731 251 746" data-label="Text"><p>nodes</p></div><div data-bbox="230 761 586 777" data-label="Text"><p><b>{HTMLElement, HTMLElementArray}</b> 指定节点。</p></div><div data-bbox="196 791 268 805" data-label="Text"><p>callback</p></div><div data-bbox="230 819 692 836" data-label="Text"><p><b>{Function}</b> 处理函数（函数中的 <b>this</b> 指向正在处理的节点）。</p></div><div data-bbox="196 851 237 866" data-label="Text"><p>args</p></div><div data-bbox="230 880 380 896" data-label="Text"><p><b>{Array}</b> 函数参数。</p></div>
```

返回值

`{HTMLElement, HTMLElementArray}` 指定节点。

代码示例

```
<div>1</div>
<div>2</div>
<script type="text/javascript">
// 把所有 div 的 innerHTML 设为 test
$.dom.eachNode(document.getElementsByTagName("div"), function() {
    this.innerHTML = "test";
});
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 354 374 381" data-label="Section-Header"><h1>样式控制——$.style</h1></div><div data-bbox="126 402 432 424" data-label="Section-Header"><h2>$.style.addCss(elems, css)</h2></div><div data-bbox="161 444 248 463" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 477 426 493" data-label="Text"><p>为指定 HTML 元素添加样式。</p></div><div data-bbox="161 508 248 526" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 543 250 558" data-label="Text"><p>elems</p></div><div data-bbox="230 571 704 588" data-label="Text"><p><code>{HTMLElement, HTMLElementArray, HtmlCollection}</code> 指定元素。</p></div><div data-bbox="196 602 230 616" data-label="Text"><p>css</p></div><div data-bbox="230 630 666 648" data-label="Text"><p><code>{String}</code> 类名或嵌入样式字符串，多个类名用空格隔开。</p></div><div data-bbox="161 660 229 679" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="196 693 668 711" data-label="Text"><p><code>{HTMLElement, HTMLElementArray, HtmlCollection}</code> 指定元素。</p></div><div data-bbox="161 724 248 743" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="196 757 777 845" data-label="Text"><pre>&lt;div id="test"&gt;&lt;/div&gt;
&lt;script type="text/javascript"&gt;<![CDATA[
$.style.addCss($("#test"), "background:#ff0000;width:30px;height:30px;");
//]]&gt;&lt;/script&gt;</pre></div>
```

\$.style.removeCss(elems, css)

函数说明

为指定 HTML 元素删除样式。

参数说明

elems

`{HTMLElement, HTMLElementArray, HtmlCollection}` 指定元素。

css

`{String}` 类名或嵌入样式字符串，多个类名用空格隔开。

返回值

`{HTMLElement, HTMLElementArray, HtmlCollection}` 指定元素。

代码示例

```
<div id="test"></div>
<script type="text/javascript">
//<![CDATA[
$.style.removeCss($("#test"), "background;width;height;");
//]]>
</script>
```

\$.style.getCurrentStyle(nodes, styleName[, w])

函数说明

获取指定元素的当前样式。如果指定元素有多个，则返回第一个元素的样式值。

参数说明

nodes

`{HTMLElement, HTMLElementArray, HtmlCollection}` 指定元素。

styleName

`{String}` 样式名。

w

`{Object}` 元素所在的页面的 window 对象，默认为当前 window 对象。

返回值

`{String}` 样式值。

代码示例

```
<div id="test"></div>
<script type="text/javascript">
alert($.style.getCurrentStyle($("#test"), "display"));    // outputs "block"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 195 385 220" data-label="Section-Header"><h1>事件控制——$.event</h1></div><div data-bbox="126 243 277 264" data-label="Section-Header"><h2>$.event.fix(e)</h2></div><div data-bbox="161 285 248 303" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 317 369 334" data-label="Text"><p>处理事件对象兼容性。</p></div><div data-bbox="161 348 248 367" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 383 213 397" data-label="Text"><p>e</p></div><div data-bbox="230 411 388 428" data-label="Text"><p><code>{Object}</code> 事件对象。</p></div><div data-bbox="161 441 229 460" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="196 475 494 491" data-label="Text"><p><code>{Object}</code> 经过兼容性处理的事件对象。</p></div><div data-bbox="161 504 248 523" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="196 537 550 698" 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 = $.event.fix(e || window.event);
    alert(e.pageX);
};
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="126 720 707 741" data-label="Section-Header"><h2>$.event.addEvent(elems, eventName, handler, data)</h2></div><div data-bbox="161 762 248 780" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 794 497 811" data-label="Text"><p>为指定 HTML 元素添加事件处理函数。</p></div><div data-bbox="161 824 248 843" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 859 250 875" data-label="Text"><p>elems</p></div><div data-bbox="230 887 704 906" data-label="Text"><p><code>{HTMLElement, HTMLElementArray, HtmlCollection}</code> 指定元素。</p></div><div data-bbox="196 917 302 933" data-label="Text"><p>eventName s</p></div>
```

{String} 事件名，多个事件用逗号隔开。

handler

{Function} 事件委托函数。

data

{Mixed} 额外传入的数据。

返回值

{HtmlElement, HtmlElementArray, HtmlCollection} 指定元素。

代码示例

```
<input type="button" id="test" value="test" />
<script type="text/javascript">
$.event.addEvent($("#test"), "click", function() {
    alert(this.value);
});
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 459 688 480" data-label="Section-Header"><h2><b>$.event.removeEvent(elems, eventName, handler)</b></h2></div><div data-bbox="160 501 248 519" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 534 497 550" data-label="Text"><p>为指定 HTML 元素移除事件委托函数。</p></div><div data-bbox="160 564 248 584" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 599 250 614" data-label="Text"><p>elems</p></div><div data-bbox="230 627 704 644" data-label="Text"><p><b>{HtmlElement, HtmlElementArray, HtmlCollection}</b> 指定元素。</p></div><div data-bbox="195 657 302 673" data-label="Text"><p>eventNames</p></div><div data-bbox="230 687 542 704" data-label="Text"><p><b>{String}</b> 事件名，多个事件用逗号隔开。</p></div><div data-bbox="195 717 263 732" data-label="Text"><p>handler</p></div><div data-bbox="230 746 440 763" data-label="Text"><p><b>{Function}</b> 事件委托函数。</p></div><div data-bbox="195 776 238 792" data-label="Text"><p>data</p></div><div data-bbox="230 805 438 822" data-label="Text"><p><b>{Mixed}</b> 额外传入的数据。</p></div><div data-bbox="160 835 229 854" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="195 868 668 885" data-label="Text"><p><b>{HtmlElement, HtmlElementArray, HtmlCollection}</b> 指定元素。</p></div><div data-bbox="160 897 248 917" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="195 931 549 949" data-label="Text"><pre>&lt;input type="button" id="test" value="test" /&gt;</pre></div>
```



```
<script type="text/javascript">
var eventFn = function() { alert(this.value); };
$.event.addEvent($("#test"), "click", eventFn);
$.event.removeEvent($("#test"), "click", eventFn);
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 188 454 214" data-label="Section-Header"><h2>浏览器判断——$.browser</h2></div><div data-bbox="125 236 245 256" data-label="Section-Header"><h3>$.browser</h3></div><div data-bbox="160 277 248 296" data-label="Section-Header"><h4>对象说明</h4></div><div data-bbox="195 310 457 326" data-label="Text"><p>获取当前浏览器的类型以及版本。</p></div><div data-bbox="160 340 248 359" data-label="Section-Header"><h4>属性说明</h4></div><div data-bbox="195 375 260 390" data-label="Text"><p>version</p></div><div data-bbox="229 403 419 421" data-label="Text"><p><b>{String}</b> 浏览器版本号。</p></div><div data-bbox="195 434 247 449" data-label="Text"><p>safari</p></div><div data-bbox="229 463 472 480" data-label="Text"><p><b>{Boolean}</b> 是否 safari 浏览器。</p></div><div data-bbox="195 494 248 510" data-label="Text"><p>opera</p></div><div data-bbox="229 522 474 540" data-label="Text"><p><b>{Boolean}</b> 是否 opera 浏览器。</p></div><div data-bbox="195 553 240 568" data-label="Text"><p>msie</p></div><div data-bbox="229 581 445 599" data-label="Text"><p><b>{Boolean}</b> 是否 IE 浏览器。</p></div><div data-bbox="195 611 256 627" data-label="Text"><p>mozilla</p></div><div data-bbox="229 640 483 658" data-label="Text"><p><b>{Boolean}</b> 是否 Mozilla 浏览器。</p></div><div data-bbox="160 670 248 689" data-label="Section-Header"><h4>代码示例</h4></div><div data-bbox="195 704 528 817" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[
if ($.browser.msie) {
    alert($.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} XMLHttpRequest 对象。

代码示例

```
<script type="text/javascript"><br/>var xhr = $.ajax.createXhr();<br/>//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="126 391 628 412" data-label="Section-Header"><h2>$.ajax.send(url[, method, data, options, xhr])</h2></div><div data-bbox="162 433 248 451" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="197 466 324 483" data-label="Text"><p>发送 Ajax 请求。</p></div><div data-bbox="161 497 248 515" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="197 531 224 546" data-label="Text"><p>url</p></div><div data-bbox="230 560 384 577" data-label="Text"><p><b>{String}</b> 请求地址。</p></div><div data-bbox="197 590 262 605" data-label="Text"><p>method</p></div><div data-bbox="230 619 647 636" data-label="Text"><p><b>{String}</b> 发送方式, "POST"或"GET", 默认为"GET"。</p></div><div data-bbox="197 649 238 664" data-label="Text"><p>data</p></div><div data-bbox="230 679 407 696" data-label="Text"><p><b>{Object}</b> 发送的数据。</p></div><div data-bbox="197 709 260 725" data-label="Text"><p>options</p></div><div data-bbox="230 737 547 754" data-label="Text"><p><b>{Object}</b> 其他可选参数。常用参数包括：</p></div><div data-bbox="230 767 380 783" data-label="Text"><p>headers: 请求头；</p></div><div data-bbox="230 797 391 813" data-label="Text"><p>timeout: 超时时间；</p></div><div data-bbox="230 826 491 842" data-label="Text"><p>onSuccess: 请求成功后的操作；</p></div><div data-bbox="230 855 464 872" data-label="Text"><p>onError: 请求失败时的操作；</p></div><div data-bbox="230 885 489 901" data-label="Text"><p>onTimeout: 请求超时时的操作。</p></div><div data-bbox="197 915 227 931" data-label="Text"><p>xhr</p></div>
```

`{XmlHttpRequest}` 用于发送请求的 `XmlHttpRequest` 对象，如不指定则自动创建。

返回值

`{XmlHttpRequest}` 进行本次 Ajax 请求的 `XMLHttpRequest` 对象。

代码示例

```
<script type="text/javascript">
$.ajax.send(
    "ajaxtest.asp",
    "POST",
    { username : "tom", password : "iamtom" },
    {
        headers : { "Content-Type" : "text/html;charset=GBK" },    // 指定为 GBK 编码
        timeout : 30000,      // 超时时间 30 秒
        onSuccess : function(xhr) {
            alert(xhr.responseText);    // 请求成功后输出返回的文本
        },
        onTimeout : function() {
            alert("timeout");
        },
        onError : function(xhr) {
            alert("error");
        }
    }
);
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="126 677 666 699" data-label="Section-Header"><h2><b>$.ajax.importJs(url[, onComplete, charset, doc])</b></h2></div><div data-bbox="161 719 248 738" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 751 440 769" data-label="Text"><p>动态加载外部 Javascript 文件。</p></div><div data-bbox="161 781 248 801" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 815 224 832" data-label="Text"><p>url</p></div><div data-bbox="230 845 344 863" data-label="Text"><p><code>{String}</code> URL。</p></div><div data-bbox="195 874 299 892" data-label="Text"><p>onComplete</p></div><div data-bbox="230 903 546 922" data-label="Text"><p><code>{Function}</code> 加载完成后执行的回调函数。</p></div>
```

charset

`{String}` 文件编码。

doc

`{HtmlDocument}` 文档对象，默认为当前文档。

代码示例

```
<script type="text/javascript">
$.ajax.importJs("test.js", function() {
    alert("loaded");
}, "gb2312");
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 339 450 364" data-label="Section-Header"><h2>Cookie 操作——$.cookie</h2></div><div data-bbox="126 387 326 407" data-label="Section-Header"><h3>$.cookie.encoder</h3></div><div data-bbox="161 429 248 447" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="196 463 688 480" data-label="Text"><p>cookie 编码函数（可设置项），默认为 <code>encodeURIComponent</code>。</p></div><div data-bbox="126 502 326 521" data-label="Section-Header"><h3>$.cookie.decoder</h3></div><div data-bbox="161 544 248 562" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="196 578 688 595" data-label="Text"><p>cookie 解码函数（可设置项），默认为 <code>decodeURIComponent</code>。</p></div><div data-bbox="126 616 347 639" data-label="Section-Header"><h3>$.cookie.get(name)</h3></div><div data-bbox="161 658 248 677" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="196 692 328 709" data-label="Text"><p>获取 Cookie 值。</p></div><div data-bbox="161 723 248 741" data-label="Section-Header"><h4>参数说明</h4></div><div data-bbox="196 757 247 771" data-label="Text"><p>name</p></div><div data-bbox="230 785 387 802" data-label="Text"><p><code>{String}</code> Cookie 名。</p></div><div data-bbox="161 815 229 835" data-label="Section-Header"><h4>返回值</h4></div><div data-bbox="196 849 351 867" data-label="Text"><p><code>{String}</code> Cookie 值。</p></div><div data-bbox="161 879 248 898" data-label="Section-Header"><h3>代码示例</h3></div><div data-bbox="196 911 527 929" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[</pre></div>
```

```
alert($.cookie.get("userName"));
//]]></script>
```

\$.cookie.set(name, value, [expires, domain, path, secure])

函数说明

设置 Cookie 值。

参数说明

name

{String} Cookie 名。

value

{Mixed} Cookie 值。

expire

{Number,Date,String} 过期时间。

数字型：单位为秒；

日期型；

字符串：由数字和单位组成的字符串，如 "1y"，其中 y 是年，M 是月，d 是日，h 是小时，m 是月，s 是秒。

domain

{String} 域。

path

{String} 路径。

secure

{Boolean} 是否仅把 Cookie 发送给受保护的服务器（https）。

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>
```

\$.cookie.del(name[, domain, path])

函数说明

删除 Cookie。

参数说明

name

{String} Cookie 名。

domain

{String} 域。

path

{String} 路径。

代码示例

```
<script type="text/javascript">
$.cookie.set("userName", "user");
$.cookie.del("userName");
alert($.cookie.get("userName"));    // outputs ""
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 539 283 565" data-label="Section-Header"><h2>String 类扩展</h2></div><div data-bbox="125 587 428 609" data-label="Section-Header"><h3><b>String.prototype.trim([str])</b></h3></div><div data-bbox="161 629 248 648" data-label="Section-Header"><h4>函数说明</h4></div><div data-bbox="196 662 477 679" data-label="Text"><p>去掉当前字符串两端的某段字符串。</p></div><div data-bbox="161 692 248 711" data-label="Section-Header"><h4>参数说明</h4></div><div data-bbox="196 727 224 742" data-label="Text"><p>str</p></div><div data-bbox="230 756 544 773" data-label="Text"><p><b>{String}</b> 要去掉的字符串，默认为空白。</p></div><div data-bbox="161 786 229 805" data-label="Section-Header"><h4>返回值</h4></div><div data-bbox="196 818 335 835" data-label="Text"><p>修整后的字符串。</p></div><div data-bbox="161 848 248 868" data-label="Section-Header"><h4>代码示例</h4></div><div data-bbox="196 882 528 948" data-label="Text"><pre>&lt;input type="text" id="userName" /&gt;
&lt;input type="button" value="Check" /&gt;
&lt;script type="text/javascript"&gt;<![CDATA[</pre></div>
```

```
$("#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");  
    }  
});  
//]]></script>
```

String.prototype.left(n)

函数说明

从左边开始截取一定长度的子字符串。

参数说明

n

{Number} 子字符串长度。

返回值

{String} 子字符串。

代码示例

```
<script type="text/javascript"><br/>var testStr = "abcd";<br/>alert(testStr.left(2)); // outputs "ab"<br/>//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 618 405 641" data-label="Section-Header"><h2>String.prototype.right(n)</h2></div><div data-bbox="161 661 248 680" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 693 493 711" data-label="Text"><p>从右边开始截取一定长度的子字符串。</p></div><div data-bbox="161 724 248 743" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 759 214 773" data-label="Text"><p>n</p></div><div data-bbox="229 787 436 805" data-label="Text"><p><b>{Number}</b> 子字符串长度。</p></div><div data-bbox="161 817 229 837" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="196 850 348 868" data-label="Text"><p><b>{String}</b> 子字符串。</p></div><div data-bbox="161 880 248 900" data-label="Section-Header"><h3>代码示例</h3></div><div data-bbox="194 912 527 931" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[</pre></div>
```

```
var testStr = "abcd";
alert(testStr.right(2));      // outputs "cd"
//]]></script>
```

String.prototype.has(str[, splitter])

函数说明

检查当前字符串中是否存在以指定分隔符隔开的一段子字符串。

参数说明

str

{String} 子字符串。

splitter

{String} 分隔符，如果省略此参数则效果同 `indexOf` 方法。

返回值

{Boolean} 是否包含子字符串。

代码示例

```
<script type="text/javascript">
var testStr1 = "ab cd", test2 = "abcd";
alert(testStr1.has("cd", " "));      // outputs "true"
alert(testStr2.has("cd", " "));      // outputs "false"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 630 549 652" data-label="Section-Header"><h2>String.format(str, arg1, arg2, ..., argN)</h2></div><div data-bbox="161 672 248 690" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 705 317 721" data-label="Text"><p>格式化字符串。</p></div><div data-bbox="161 736 248 755" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 770 223 785" data-label="Text"><p>str</p></div><div data-bbox="230 798 454 816" data-label="Text"><p><b>{String}</b> 要格式化的字符串。</p></div><div data-bbox="195 828 301 845" data-label="Text"><p>argN (N &gt; 0)</p></div><div data-bbox="230 857 359 875" data-label="Text"><p><b>{Mixed}</b> 参数值</p></div><div data-bbox="161 887 229 906" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="195 920 420 938" data-label="Text"><p><b>{String}</b> 格式化后的字符串。</p></div>
```


代码示例

```
<script type="text/javascript">
alert(String.format("%1, %2, %3", "a", "b", "c")); // outputs "a, b, c"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="90 172 275 198" data-label="Section-Header"><h1>Array 类扩展</h1></div><div data-bbox="125 219 594 241" data-label="Section-Header"><h2>Array.prototype.indexOf(elem[, startPos])</h2></div><div data-bbox="161 262 248 280" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 294 423 310" data-label="Text"><p>在当前数组中检索指定元素。</p></div><div data-bbox="161 324 248 343" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 358 241 373" data-label="Text"><p>elem</p></div><div data-bbox="229 388 384 404" data-label="Text"><p><b>{Mixed}</b> 指定元素。</p></div><div data-bbox="196 417 267 432" data-label="Text"><p>startPos</p></div><div data-bbox="229 446 590 463" data-label="Text"><p><b>{Number}</b> 开始检索的位置，0 是第一个位置。</p></div><div data-bbox="161 477 229 495" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="196 510 898 527" data-label="Text"><p><b>{Number}</b> 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。</p></div><div data-bbox="161 540 248 559" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="196 573 527 662" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[
var arr = [1, 2, 3, 4, 1];
alert(arr.indexOf(1, 2)); // outputs "4"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="125 683 428 704" data-label="Section-Header"><h2>Array.prototype.remove(n)</h2></div><div data-bbox="161 725 248 744" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 758 440 774" data-label="Text"><p>删除当前数组指定位置的元素。</p></div><div data-bbox="161 788 248 807" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 822 214 837" data-label="Text"><p>n</p></div><div data-bbox="229 851 400 869" data-label="Text"><p><b>{Number}</b> 指定位置。</p></div><div data-bbox="161 881 229 900" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="196 913 344 932" data-label="Text"><p><b>{Array}</b> 当前数组。</p></div>
```

代码示例

```
<script type="text/javascript">
var arr = [1, 2, 3, 4, 1];
alert(arr.remove(0)); // outputs "2,3,4,1"
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 196 322 220" data-label="Section-Header"><h1>Function 类扩展</h1></div><div data-bbox="126 243 694 265" data-label="Section-Header"><h2>Function.prototype.bind(obj[, arg1, arg2, ..., argN])</h2></div><div data-bbox="161 285 248 304" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 317 387 335" data-label="Text"><p>为函数绑定 this 和参数。</p></div><div data-bbox="161 348 248 367" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="196 382 227 399" data-label="Text"><p>obj</p></div><div data-bbox="230 411 457 429" data-label="Text"><p><b>{Mixed}</b> 指定为 this 的对象。</p></div><div data-bbox="196 441 301 458" data-label="Text"><p>argN (N &gt; 0)</p></div><div data-bbox="230 471 349 487" data-label="Text"><p><b>{Mixed}</b> 参数。</p></div><div data-bbox="161 500 229 519" data-label="Section-Header"><h3>返回值</h3></div><div data-bbox="196 534 494 552" data-label="Text"><p><b>{Function}</b> 绑定了 this 和参数的函数。</p></div><div data-bbox="161 564 248 583" data-label="Section-Header"><h2>代码示例</h2></div><div data-bbox="196 597 627 758" data-label="Text"><pre>&lt;script type="text/javascript"&gt;<![CDATA[
function test(arg) {
    alert(this.value + arg);
}
var obj = { value : 1 };
test.bind(obj, 2)();    // outputs "3" (obj.value + 2 = 3)
//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="91 776 261 801" data-label="Section-Header"><h1>Date 类扩展</h1></div><div data-bbox="126 823 253 846" data-label="Section-Header"><h2>Date.now()</h2></div><div data-bbox="161 865 248 885" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="196 898 405 916" data-label="Text"><p>获取当前日期的毫秒表示。</p></div>
```

返回值

{Number} 当前日期的毫秒表示。

代码示例

```
<script type="text/javascript"><br/>alert(Date.now());<br/>//]]&gt;&lt;/script&gt;</pre></div><div data-bbox="126 237 503 259" data-label="Section-Header"><h2>Date.prototype.format(formation)</h2></div><div data-bbox="160 279 248 298" data-label="Section-Header"><h3>函数说明</h3></div><div data-bbox="195 310 528 328" data-label="Text"><p>把当前时间对象转换为指定格式的字符串。</p></div><div data-bbox="160 342 248 362" data-label="Section-Header"><h3>参数说明</h3></div><div data-bbox="195 376 277 392" data-label="Text"><p>formation</p></div><div data-bbox="229 404 348 423" data-label="Text"><p><b>{String}</b> 格式。</p></div><div data-bbox="229 434 361 452" data-label="Text"><p><b>yyyy</b>: 4 位年份</p></div><div data-bbox="229 464 343 482" data-label="Text"><p><b>yy</b>: 2 位年份</p></div><div data-bbox="229 493 353 511" data-label="Text"><p><b>MM</b>: 2 位月份</p></div><div data-bbox="229 522 338 541" data-label="Text"><p><b>M</b>: 1 位月份</p></div><div data-bbox="229 552 344 570" data-label="Text"><p><b>dd</b>: 2 位日数</p></div><div data-bbox="229 582 334 600" data-label="Text"><p><b>d</b>: 1 位日数</p></div><div data-bbox="229 611 447 629" data-label="Text"><p><b>HH</b>: 24 小时制 2 位小时数</p></div><div data-bbox="229 641 435 659" data-label="Text"><p><b>H</b>: 24 小时制 1 位小时数</p></div><div data-bbox="229 671 444 689" data-label="Text"><p><b>hh</b>: 12 小时制 2 位小时数</p></div><div data-bbox="229 700 433 719" data-label="Text"><p><b>h</b>: 12 小时制 1 位小时数</p></div><div data-bbox="229 730 372 748" data-label="Text"><p><b>mm</b>: 2 位分钟数</p></div><div data-bbox="229 759 357 777" data-label="Text"><p><b>m</b>: 1 位分钟数</p></div><div data-bbox="229 789 343 807" data-label="Text"><p><b>ss</b>: 2 位秒数</p></div><div data-bbox="229 818 334 837" data-label="Text"><p><b>s</b>: 1 位秒数</p></div><div data-bbox="160 849 230 869" data-label="Section-Header"><h2>返回值</h2></div><div data-bbox="195 881 454 901" data-label="Text"><p><b>{String}</b> 指定格式的日期字符串。</p></div>
```

代码示例

```
<script type="text/javascript"><br/>var date = new Date(2009, 9, 1);<br/>alert(date.format("yyyy-MM-dd")); // outputs "2009-10-01"<br/>//]]&gt;&lt;/script&gt;</pre></div>
```