

## 语法

通过选取 HTML 元素，并对选取的元素执行某些操作。

```
$(selector).action()
```

- 美元符号定义 jQuery
- 选择符 ( selector ) "查询"和"查找" HTML 元素
- jQuery 的 action() 执行对元素的操作

eg:

```
$(this).hide() - 隐藏当前元素
```

```
$("p").hide() - 隐藏所有段落
```

```
$("p.test").hide() - 隐藏所有 class="test" 的段落
```

```
$("#test").hide() - 隐藏所有 id="test" 的元素
```

## 文档就绪事件

为了防止文档在完全加载（就绪）之前运行 jQuery 代码。在文档没有完全加载之前就运行函数，操作可能失败。

```
$(document).ready(function() {  
    // 开始写 jQuery 代码...  
});
```

简洁写法:

```
$(function() {  
    // 开始写 jQuery 代码...  
});
```

## 选择器

jQuery 元素选择器 `$()`: 基于元素名选取元素。eg. `$("p")`

jQuery #id 选择器 `$("#id")`: 通过 HTML 元素的 id 属性选取指定的元素。eg. `$("#test")`

jQuery 类选择器 `$(".class")`: 通过指定的 class 查找元素。eg. `$(".test")`

jQuery CSS 选择器: 用于改变 HTML 元素的 CSS 属性。eg. `$("p").css("background-color", "red");`

语法	描述
<code>\$("*")</code>	选取所有元素
<code>\$(this)</code>	选取当前 HTML 元素
<code>\$("p.intro")</code>	选取 class 为 intro 的 <p> 元素
<code>\$("p:first")</code>	选取第一个 <p> 元素
<code>\$("ul li:first")</code>	选取第一个 <ul> 元素的第一个 <li> 元素
<code>\$("ul li:first-child")</code>	选取每个 <ul> 元素的第一个 <li> 元素
<code>\$("[href]")</code>	选取带有 href 属性的元素
<code>\$("a[target='_blank']")</code>	选取所有 target 属性值等于 "_blank" 的 <a> 元素
<code>\$("a[target!='_blank']")</code>	选取所有 target 属性值不等于 "_blank" 的 <a> 元素
<code>\$(":button")</code>	选取所有 type="button" 的 <input> 元素 和 <button> 元素

<code>\$("tr:even")</code>	选取偶数位置的 <code>&lt;tr&gt;</code> 元素
<code>\$("tr:odd")</code>	选取奇数位置的 <code>&lt;tr&gt;</code> 元素

## 独立文件中使用 jQuery 函数

如果您的网站包含许多页面，并且您希望您的 jQuery 函数易于维护，那么请把您的 jQuery 函数放到独立的 .js 文件中。

当我们在教程中演示 jQuery 时，会将函数直接添加到 `<head>` 部分中。不过，把它们放到一个单独的文件中会更好，就像这样（通过 `src` 属性来引用文件）：

### 实例

```
<head>
<scriptsrc="http://ajax.googleapis.com/ajax/libs/jquery/1.10.2/jquery.min.js">
<script>
<script src="my_jquery_functions.js">
</script>
</head>
```

## 事件

大多数 [DOM 事件](#) 都有一个等效的 jQuery 方法。常见 DOM 事件：

鼠标事件	键盘事件	表单事件	文档/窗口事件
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

eg.

```
$("p").click(function() {
    // action goes here!!
});
```

常见的 jQuery 事件方法：

`$(document).ready()`

允许我们在文档完全加载完后执行函数。

`click()`

当按钮点击事件被触发时会调用一个函数。在用户点击 HTML 元素时执行。

`dblclick()`

当双击元素时，会发生 `dblclick` 事件。

`mouseenter()`

当鼠标指针穿过元素时，会发生 `mouseenter` 事件。

`mouseleave()`

当鼠标指针离开元素时，会发生 `mouseleave` 事件。

`mousedown()`

当鼠标指针移动到元素上方，并按下鼠标按键时，会发生 `mousedown` 事件。

`mouseup()`

当在元素上松开鼠标按钮时，会发生 `mouseup` 事件。

## hover()

hover() 方法用于模拟光标悬停事件。

当鼠标移动到元素上时，会触发指定的第一个函数(mouseenter);当鼠标移出这个元素时，会触发指定的第二个函数(mouseleave)。

```
$("#p1").hover(function(){
    alert("You entered p1!");
},
function(){
    alert("Bye! You now leave p1!");
});
```

## focus()

当元素获得焦点时，发生 focus 事件。当通过鼠标点击选中元素或通过 tab 键定位到元素时，该元素就会获得焦点。

## blur()

当元素失去焦点时，发生 blur 事件。

比较keypress、keydown与keyup

- **keydown**：在键盘上按下某键时发生，一直按着则会不断触发（opera浏览器除外），它返回的是键盘代码;
- **keypress**：在键盘上按下一个按键，并产生一个**字符**时发生, 返回ASCII码。注意: shift、alt、ctrl 等键按下并不会产生字符，所以监听无效。
- **keyup**：用户松开某一个按键时触发，与keydown相对，返回键盘代码。