

# JQUERY

**jQuery** 是专注于简化 DOM 操作, AJAX 调用和 Event 处理的 JavaScript 库。它被 JavaScript 开发者频繁使用。

jQuery 使用 `$(selector).action()` 的格式给一个 (或多个) 元素绑定事件。具体来说, `$(selector)` 调用 jQuery 函数选择 `selector` 元素, 并给它绑定一个叫 `.action` 的事件 API。

```
$(document).ready(function(){  
    alert("Hello World!");  
    $("#blackBox").hide();  
});
```

上述代码和以下代码功能相同:

```
window.onload = function() {  
    alert( "Hello World!" );  
    document.getElementById("blackBox").style.display = "none";  
};
```

---

<https://zh.wikipedia.org/wiki/JQuery>

[Jquery维基百科--链接](#)

# jQuery 语法

通过 jQuery，您可以选取（查询，query）HTML 元素，并对它们执行“操作”（actions）。

## jQuery 语法

jQuery 语法是通过选取 HTML 元素，并对选取的元素执行某些操作。

基础语法： `$(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

实例：

- `$(this).hide()` - 隐藏当前元素
- `$("p").hide()` - 隐藏所有 `<p>` 元素
- `$("#p.test").hide()` - 隐藏所有 `class="test"` 的 `<p>` 元素
- `$("#test").hide()` - 隐藏所有 `id="test"` 的元素



你对 CSS 选择器熟悉吗？

jQuery 使用的语法是 XPath 与 CSS 选择器语法的组合。在本教程接下来的章节，您将学习到更多有关选择器的语法。

## 文档就绪事件

您也许已经注意到在我们的实例中的所有 jQuery 函数位于一个 `document ready` 函数中：

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

这是为了防止文档在完全加载（就绪）之前运行 jQuery 代码，即在 DOM 加载完成后才可以对 DOM 进行操作。

提示：简洁写法（与以上写法效果相同）：

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

以上两种方式你可以选择你喜欢的方式实现文档就绪后执行 jQuery 方法。

jQuery 入口函数:

```
$(document).ready(function(){  
    // 执行代码  
});  
或者  
$(function(){  
    // 执行代码  
});
```

JavaScript 入口函数:

```
window.onload = function () {  
    // 执行代码  
}
```

jQuery 入口函数与 JavaScript 入口函数的区别:

- jQuery 的入口函数是在 html 所有标签(DOM)都加载之后, 就会去执行。
- JavaScript 的 window.onload 事件是等到所有内容, 包括外部图片之类的文件加载完后, 才会执行。

#### load和ready区别

	window.onload	\$(document).ready()
执行时机	必须等待网页全部加载完毕 (包括图片等), 然后再执行包裹代码	只需要等待网页中的DOM结构加载完毕, 就能执行包裹的代码
执行次数	只能执行一次, 如果第二次, 那么第一次的执行会被覆盖	可以执行多次, 第N次都不会被上一次覆盖
简写方案	无	<code>\$(function () {  });</code>

# jQuery 选择器

jQuery 选择器允许您对 HTML 元素组或单个元素进行操作。

## jQuery 选择器

jQuery 选择器允许您对 HTML 元素组或单个元素进行操作。

jQuery 选择器基于元素的 id、类、类型、属性、属性值等“查找”（或选择）HTML 元素。它基于已经存在的 [CSS 选择器](#)，除此之外，它还有一些自定义的选择器。

jQuery 中所有选择器都以美元符号开头：\$()。

## 元素选择器

jQuery 元素选择器基于元素名选取元素。

在页面中选取所有 <p> 元素：

```
$("p")
```

实例

用户点击按钮后，所有 <p> 元素都隐藏：

实例

```
$(document).ready(function(){
    $("button").click(function(){
        $("p").hide();
    });
});
```

## #id 选择器

jQuery #id 选择器通过 HTML 元素的 id 属性选取指定的元素。

页面中元素的 id 应该是唯一的，所以您要在页面中选取唯一的元素需要通过 #id 选择器。

通过 id 选取元素语法如下：

```
$("#test")
```

实例

当用户点击按钮后，有 id="test" 属性的元素将被隐藏：

实例

```
$(document).ready(function(){
    $("button").click(function(){
        $("#test").hide();
    });
});
```

# .class 选择器

jQuery 类选择器可以通过指定的 class 查找元素。

语法如下：

```
$(".test")
```

## 实例

用户点击按钮后所有带有 class="test" 属性的元素都隐藏：

实例

```
$(document).ready(function(){
    $("button").click(function(){
        $(".test").hide();
    });
});
```

## 更多实例

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

# jQuery 事件

jQuery 是为事件处理特别设计的。

## 什么是事件？

页面对不同访问者的响应叫做事件。

事件处理程序指的是当 HTML 中发生某些事件时所调用的方法。

实例：

- 在元素上移动鼠标。
- 选取单选按钮
- 点击元素

在事件中经常使用术语“触发”（或“激发”）例如：“当您按下按键时触发 keypress 事件”。

常见 DOM 事件：

鼠标事件	键盘事件	表单事件	文档/窗口事件
<a href="#">click</a>	<a href="#">keypress</a>	<a href="#">submit</a>	<a href="#">load</a>
<a href="#">dblclick</a>	<a href="#">keydown</a>	<a href="#">change</a>	<a href="#">resize</a>
<a href="#">mouseenter</a>	<a href="#">keyup</a>	<a href="#">focus</a>	<a href="#">scroll</a>
<a href="#">mouseleave</a>		<a href="#">blur</a>	<a href="#">unload</a>
<a href="#">hover</a>			

## jQuery 事件方法语法

在 jQuery 中，大多数 DOM 事件都有一个等效的 jQuery 方法。

页面中指定一个点击事件：

```
$("#p").click();
```

下一步是定义什么时间触发事件。您可以通过一个事件函数实现：

```
$("#p").click(function(){  
    // 动作触发后执行的代码!!  
});
```



## 常用的 jQuery 事件方法

### `$(document).ready()`

`$(document).ready()` 方法允许我们在文档完全加载完后执行函数。该事件方法在 [jQuery 语法](#) 章节中已经提到过。

### `click()`

`click()` 方法是当按钮点击事件被触发时会调用一个函数。

该函数在用户点击 HTML 元素时执行。

在下面的实例中，当点击事件在某个 `<p>` 元素上触发时，隐藏当前的 `<p>` 元素：

#### 实例

```
$("#p").click(function(){  
    $(this).hide();  
});
```

尝试一下 »

### `dblclick()`

当双击元素时，会发生 `dblclick` 事件。

`dblclick()` 方法触发 `dblclick` 事件，或规定当发生 `dblclick` 事件时运行的函数：

#### 实例

```
$("#p").dblclick(function(){  
    $(this).hide();  
});
```

## mouseenter()

当鼠标指针穿过元素时，会发生 mouseenter 事件。

mouseenter() 方法触发 mouseenter 事件，或规定当发生 mouseenter 事件时运行的函数：

### 实例

```
$("#p1").mouseenter(function(){  
    alert('您的鼠标移到了 id="p1" 的元素上!');  
});
```

尝试一下 »

## mouseleave()

当鼠标指针离开元素时，会发生 mouseleave 事件。

mouseleave() 方法触发 mouseleave 事件，或规定当发生 mouseleave 事件时运行的函数：

### 实例

```
$("#p1").mouseleave(function(){  
    alert("再见，您的鼠标离开了该段落。");  
});
```

尝试一下 »

## 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。

mousedown() 方法触发 mousedown 事件，或规定当发生 mousedown 事件时运行的函数：

### 实例

```
$("#p1").mousedown(function(){  
    alert("鼠标在该段落上按下！");  
});
```

尝试一下 »



## mouseup()

当在元素上松开鼠标按钮时，会发生 mouseup 事件。

mouseup() 方法触发 mouseup 事件，或规定当发生 mouseup 事件时运行的函数：

### 实例

```
$("#p1").mouseup(function(){
    alert("鼠标在段落上松开。");
});
```

尝试一下 »

## 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)。

### 实例

```
$("#p1").hover(
    function(){
        alert("你进入了 p1!");
    },
    function(){
        alert("拜拜！现在你离开了 p1!");
    }
);
```

尝试一下 »

## 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。

focus() 方法触发 focus 事件，或规定当发生 focus 事件时运行的函数：

### 实例

```
$("#input").focus(function(){
    $(this).css("background-color", "#cccccc");
});
```

尝试一下 »

## blur()

当元素失去焦点时，发生 blur 事件。

blur() 方法触发 blur 事件，或规定当发生 blur 事件时运行的函数：

### 实例

```
$("#input").blur(function(){
    $(this).css("background-color", "#ffffff");
});
```

尝试一下 »

