

添加jQuery

1. 创建一个jQuery文件与html处于同一文件夹下，在<head>中添加<script src="jquery-3.7.1.min.js"></script>
2. 在中添加<script src="https://code.jquery.com/jquery-3.7.1.min.js"></script>

jQuery语法

基础语法: **`$(*selector)*.action*`**()

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

简写为`$(function(){})`

设置属性 - attr()

jQuery attr() 方法也用于设置/改变属性值。

下面的例子演示如何改变（设置）链接中 href 属性的值：

实例

```
$("#button").click(function(){ $("#runoob").attr("href","http://www.runoob.com/jquery"); });
```

jQuery remove() 方法

jQuery remove() 方法删除被选元素及其子元素。

实例

```
$("#div1").remove();
```

[尝试一下 »](#)

jQuery empty() 方法

jQuery empty() 方法删除被选元素的子元素。

实例

```
$("#div1").empty();
```

[尝试一下 »](#)

过滤被删除的元素

jQuery remove() 方法也可接受一个参数，允许您对被删元素进行过滤。

该参数可以是任何 jQuery 选择器的语法。

下面的例子删除 class="italic" 的所有

元素：

实例

```
$("#p").remove(".italic");
```

jQuery 获取 DOM 节点

使用选择器选取 DOM 节点

在 jQuery 中，可以使用选择器语法来选取 DOM 节点。选择器可以根据元素的标签名、类名、ID、属性等进行选取。下面是一些常用的选择器示例：

1. 根据标签名选取元素：

```
$('#p') // 选取所有 <p> 元素
```

JavaScript

Copy

2. 根据类名选取元素：

```
$('.red') // 选取所有类名为 "red" 的元素
```

JavaScript

Copy

3. 根据 ID 选取元素：

```
$('#myDiv') // 选取 ID 为 "myDiv" 的元素
```

JavaScript

Copy

4. 根据属性选取元素：

```
 $('[name="username"]') // 选取所有 name 属性为 "username" 的元素
```

JavaScript

Copy

在选择器选取的基础上，还可以使用链式调用的方式进行进一步操作。

使用 `.get()` 方法获取 DOM 节点

除了常规的选择器方法，jQuery 还提供了 `.get()` 方法来获取选取的 DOM 节点。

```
var element = $('p').get(0); // 获取第一个 <p> 元素，并将其转换为 JavaScript 原生对象
```

JavaScript

Copy

使用 `.get()` 方法可以将 jQuery 对象转换为普通的 JavaScript 对象，从而直接使用 JavaScript 的 DOM API 进行操作。这在某些情况下非常有用，例如需要使用原生方法绑定事件或修改样式等。

示例代码

下面是一个示例代码，演示了如何使用 jQuery 获取 DOM 节点并进行操作：

```
<!DOCTYPE html>
<html>
<head>
  <title>jQuery Get DOM Node</title>
  <style>
    .highlight {
      background-color: yellow;
    }
  </style>
</head>
<body>
  <h1>Welcome to jQuery Get DOM Node</h1>
  <p>This is a paragraph.</p>
  <p class="highlight">This is another paragraph.</p>
  <div id="myDiv">
    <p>This is a paragraph inside a div.</p>
  </div>

  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    // 使用选择器选取 DOM 节点
    var paragraphs = ('p');

    // 遍历并修改选取的节点样式
    paragraphs.each(function() {(this).addClass('highlight');
    });

    // 使用 .get() 方法获取第一个 p 元素，并修改其文本内容
```

```
var firstParagraph = $('p').get(0);
firstParagraph.textContent = 'Modified paragraph content';
</script>
</body>
</html>
```

jQuery Fading 方法

通过 jQuery，您可以实现元素的淡入淡出效果。

jQuery 拥有下面四种 fade 方法：

- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()

jQuery 滑动方法

通过 jQuery，您可以在元素上创建滑动效果。

jQuery 拥有以下滑动方法：

- slideDown()
- slideUp()
- slideToggle()

遍历

jQuery parent() 方法

parent() 方法返回被选元素的直接父元素。

该方法只会向上一级对 DOM 树进行遍历。

下面的例子返回每个 元素的直接父元素：

实例

```
$(document).ready(function(){ $("span").parent();
```

jQuery children() 方法

children() 方法返回被选元素的所有直接子元素。

该方法只会向下一级对 DOM 树进行遍历。

下面的例子返回每个

元素的所有直接子元素：

实例

```
$(document).ready(function(){ $("div").children(); });
```

jQuery find() 方法

find() 方法返回被选元素的后代元素，一路向下直到最后一个后代。

下面的例子返回属于

后代的所有 元素：

实例

```
$(document).ready(function(){ $("div").find("span"); });
```

jQuery eq() 方法

eq() 方法返回被选元素中带有指定索引号的元素。

索引号从 0 开始，因此首个元素的索引号是 0 而不是 1。下面的例子选取第二个

元素（索引号 1）：

实例

```
$(document).ready(function(){ $("p").eq(1); });
```

jQuery not() 方法

not() 方法返回不匹配标准的所有元素。

提示：not() 方法与 filter() 相反。

下面的例子返回不带有类名 "url" 的所有

元素：

实例

```
$(document).ready(function(){ $("p").not(".url"); });
```

jQuery排序

```
.sort(function (a, b) {  
    return $(b).data('clicknumber') - $(a).data('clicknumber');
```

```
nums = nums.sort(function(a,b){  
    return a-b;  
});
```