

# jQuery

---

主讲：崔译

---

## 一、简介

---

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

jQuery 是一个快速、轻量级、可扩展的 js 库，它提供了易于使用的跨浏览器的API，使得访问dom，时间处理、动画效果、ajax请求变得简单。

简化了JS对DOM的操作

## 二、jQuery获取方式

---

[jQuery官网下载链接](#)

### 1、关于生产版（min版）和开发板

#### 1. 生产版

compressed, production jQuery 压缩后的。用于生产环境的版本  
去掉了所有的 不影响使用的代码和空格、换行等等  
保证jQuery文件的最小

```
function some(username,password){  
    console.log(username,password);  
}  
  
function a(b,c){console.log(b,c)}
```

#### 2. 开发板

项目开发过程中使用的版本，代码是可读的

## 三、jQuery核心

---

### 1、jQuery核心对象

```
window.jQuery = window.$ = jQuery;
```

- 在window对象中，多了两个属性，叫做jQuery 和 `$`
- `jQuery` 属性和 `$` 可以相互替代

```
// 例如jQuery中存在一个方法叫做each
window.jQuery.each()
jQuery.each()
window.$.each()
$.each()
```

## 2、关于jQuery对象

使用jQuery获取的元素是jQuery对象，不是DOM对象，不能使用DOM对象方法。

```
let div = document.getElementById("d");
div.style.color="red";
// 变量名任意，个人习惯，jQuery对象以$开头
let $div = $("#d");
//错误
$div.style.color="red";
//正确
$div.css("color", "red");
```

DOM对象可以和jQuery对象相互转换

```
//DOM --> jquery
let $div= $(dom对象);
//jquery ---> DOM
let div = $div[0]
```

## 3、注意点

### 3-1 关于赋值操作

任何jQuery 元素对象的赋值操作，基本上都是通过方法的第二个参数赋值，不会出现 `=`

```
// 错误代码
$.css() = "20px";
```

### 3-2 `$ is not defined` 错误

- jquery 的js文件没有导入
- jquery 的js文件路径错误
- jquery 的js文件导入位置错误

## 四、HelloWorld

```
<!DOCTYPE html>

<html lang="en">
```

```
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/jquery-3.3.1.js"></script>
  <script>
    // window.onload = function(){}
    $(function(){
      let $div = $(".d");
      console.log($div.html());
      console.log($div[0].innerHTML);
    });
  </script>
</head>
<body>
  <div class="d">HelloWorld</div>
</body>
</html>
```

## 五、访问节点

### 1、基本选择器

基本语法：`$(选择器)`

选择器语法 和 CSS 选择器语法一致

id / 类 / 标签 / 子元素 / 后代 / 相邻兄弟 / 属性 / 伪类

### 2、后代选择器

- `$("div>a")` `$("div a")`
- `$.find(选择器)` 在某个元素的后代中查找
- `$.children(选择器)` 选择器可以不写，默认找所有子元素，否则找满足选择器的子元素

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/jquery-3.3.1.js"></script>
  <script>
    $(function(){
      // 后代中的h1
      let h1s = $("#d").find("h1");
      console.log(h1s);
      // 所有的子元素
      h1s= $("#d").children();
      console.log(h1s);
      // 子元素中的h1
      h1s= $("#d").children("h1");
      console.log(h1s);
    });
  </script>
</head>
<body>
  <div>
    <h1>HelloWorld</h1>
  </div>
</body>
</html>
```

```
    })

    </script>
</head>
<body>

    <div id="d">
        <h1>aaa</h1>
        <h1>bbb</h1>
        <div>
            <h1>ccc</h1>
        </div>
    </div>

</body>
</html>
```

### 3、父选择器

`$.parent()` 找某个节点的 父节点

### 4、兄弟节点

方法	作用
<code>\$.siblings()</code>	所有兄弟节点
<code>\$.next()</code>	下一个兄弟节点
<code>\$.prev()</code>	上一个兄弟节点
<code>\$.nextAll()</code>	下面的所有兄弟节点
<code>\$.prevAll()</code>	上面的所有兄弟节点
<code>\$.nextUntil(selector)</code>	介于两个节点之间的所有节点（向下找）
<code>\$.prevUntil(selector)</code>	介于两个节点之间的所有节点（向上找）

### 5、`:` 选择器

方法	作用
<code>\$("span:first")</code>	选择第一个span
<code>\$("span:last")</code>	选择最后一个span
<code>\$("span:even")</code>	第偶数个span
<code>\$("span:odd")</code>	第奇数个span
<code>\$("input:button")</code> 或者 <code>\$(":button")</code>	所有的input 并且type是button的元素
<code>\$(":checkbox")</code>	所有的 复选框
<code>\$(":radio")</code>	所有的 单选按钮
<code>\$(":checked")</code>	所有选中的checkbox / radio
<code>\$(":selected")</code>	所有选中的select>option

## 6、选择器过滤

方法	作用
<code>\$.first()</code>	满足选择器的第一个元素
<code>\$.last()</code>	满足选择器的最后一个元素
<code>\$.eq(n)</code>	满足选择器的第n个元素（从0开始）
<code>\$(selector1).not(selector2)</code>	满足selector1并且不满足selector2的元素

## 7、元素的遍历

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/jquery-3.3.1.js"></script>
  <script>

    $(function () {
      // 获取所有的span
      // 控制台输出每个span的itany属性的值 jquery对象.attr("itany")
      let spans = $("span");
      // 方式1
      // for(let i = 0 ; i < spans.length ; i++)

      // {
  
```

```

        // let value = $(spans[i]).attr("itany");
        // console.log(value);
        // }

        // 方式2
        // for(let i = 0 ; i < spans.length ; i++)
        // {
        // let value = spans.eq(i).attr("itany");
        // console.log(value);
        // }

        // 方式3
        // spans.each(function(){
        // // 对于每个元素，都会执行该方法
        // // $(this) 是每次循环的元素的jquery对象
        // console.log($(this).attr("itany"));
        // $(this).css("color", "red");
        // });

        // 注意，如果对于Jquery对象数组中的所有元素进行相同的操作，可以不用遍历
        $("span").css("color", "green");
    })

</script>
</head>
<body>
    <span itany="11">1</span>
    <label for="">a</label>
    <span itany="22">2</span>
    <h1>b</h1>
    <span itany="33">3</span>
    <span itany="44">5</span>
    <div>c</div>
</body>
</html>

```

## 六、访问属性

- DOM属性
  - `$.prop("属性名")` 取值操作
  - `$.prop("属性名", "属性值")` 赋值操作
- HTML属性
  - `$.attr("属性名")` 取值操作
  - `$.attr("属性名", "属性值")` 赋值操作

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">

```

```
<title>Document</title>
<script src="js/jquery-3.3.1.js"></script>
<script>
    function select(flag)
    {
        $("checkbox").prop("checked", flag);
    }
</script>
</head>
<body>

    <input type="checkbox">
    <input type="checkbox">
    <input type="checkbox">
    <input type="checkbox">
    <input type="checkbox">

    <button onclick="select(true)">全选</button>
    <button onclick="select(false)">全不选</button>

</body>
</html>
```

## 七、访问样式

### 1、访问class

- 作为普通属性访问（第六章）
- `addClass("样式名")` 添加样式
- `removeClass("样式名")` 删除样式
- `removeClass()` 删除所有样式
- `toggleClass("样式名")` 有则删除，没有则添加

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <style>
        .content{
            width: 80%;
            margin:auto;
            padding:20px;
        }

        .header{
            width: 100%;
            height: 40px;
            border:1px solid black;
        }
    </style>
</head>
<body>
```

```

        .container{
            width: 100%;
            height: 440px;
            border:1px solid black;
        }
        .header i{
            border:1px solid black;
            border-radius:50%;
            float: right;
            font-size:20px;
            margin-top: 10px;
            vertical-align:center;
        }
        .hide{
            display:none;
        }
    </style>
    <link rel="stylesheet" href="iconfont/iconfont.css">
    <script src="js/jquery-3.3.1.js"></script>
    <script>
        function doChange()
        {
            $(".container").toggleClass("hide");
            $(".i").toggleClass('icon-down');
        }
    </script>
</head>
<body>
    <div class="content">
        <div class="header">
            <i class="iconfont icon-up" onclick="doChange()"></i>
        </div>
        <div class="container">
            content.....
        </div>
    </div>
</body>
</html>

```

## 2、访问css

- `$(".css("样式名"))` 取值操作 ( 原始css样式 background-color )
- `$(".css("样式名","样式值"))` 赋值操作
- `$(".css({}))` 一次性修改多个css样式 参数是json对象，对象属性名是css属性名，对象值是css属性值
- `$(".width())`、`$(".height())` 无参取值，有参赋值

## 八、访问内容

- `$(".html())` 无参取值，有参赋值，注意：取值取标签，赋值解析标签
- `$(".text())` 无参取值，有参赋值，注意：取值不取标签，赋值不解析标签



- `$.val()` 无参取值，有参赋值，相当于表单组件的value

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/jquery-3.3.1.js"></script>
  <script>
    function doChange()
    {
      let val = $("#source").val();
      $("#target").text( val );
      let arr = ["red", "green", "blue"];
      let color = arr[parseInt(val.length / 5)];
      if(color)
      {
        $("#target").css("color", arr[parseInt(val.length / 5)] )
      }
    }
  </script>
</head>
<body>

  <input type="text" id="source" onkeyup="doChange()">
  <h1 id="target">暂无内容</h1>

</body>
</html>
```

## 九、事件绑定

### 1、约定

事件名指的是：原始JavaScript事件，去掉on。

click mouseover blur focus .....

### 2、jquery动态事件绑定

- `$.事件名(事件处理函数)` `$("#d").click(function(){})`
- `$.on("事件名", 事件处理函数)` `$("#d").on("click", function(){})`
- `$.bind("事件名", 事件处理函数)` `$("#d").bind("click", function(){})`

### 3、解除事件绑定

`$.unbind("事件名")`

### 4、使用js触发事件

`$.事件名()`

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <script src="js/jquery-3.3.1.js"></script>
  <script>
    function doChange()
    {
      // $("#mf").submit();
      // $("#f").click();
      $("#name").focus();
    }
  </script>
</head>
<body>

  <form action="http://www.baidu.com" id="mf"></form>

  <input type="file" id="f">

  <input type="text" id="name"/>

  <button onclick="doChange()">to Baidu</button>
</body>
</html>

```

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Document</title>
  <style>
    td{
      width: 40px;
      height: 40px;
    }
    input{
      width: 208px;
      height: 40px;
      text-align:right;
      font-size:30px;
      border:1px solid black;
    }
    button{
      width: 100%;
      height: 100%;
    }
  </style>
  <script src="js/jquery-3.3.1.js"></script>
  <script>

```

```

// 使用js引擎 解释运行字符串
// let str = "1+1";
// console.log(str, eval(str) );

// str = "alert(1)";
// eval(str);

// str = '{"name":"zhangsan","age":22}';
// let obj = eval("(" + str + ")");
// console.log(obj);

// // 必须是标准JSON ,
// obj = JSON.parse(str);
// console.log(obj);

let append = true;
$(function(){
    $("button").click(function(){
        let btnText = $(this).html();
        if(btnText == "c")
        {
            $("#area").val('');
            return;
        }

        if(btnText == "=")
        {
            append = false;
            let val = $("#area").val();
            try {
                $("#area").val( eval(val) )
            } catch(e) {
                $("#area").val( "NaN" )
            }
            return;
        }

        if(!append){
            $("#area").val('');
        }

        $("#area").val( $("#area").val() + btnText );
        append = true;
    })
})

</script>
</head>

<body>
    <input type="text" id="area" value="">
    <table border="1" cellspacing="0" cellpadding="5">

        <tr>

```

```

        <td>
            <button>c</button>
        </td>
        <td>
            <button>/</button>
        </td>
        <td>
            <button>*</button>
        </td>
        <td>
            <button>-</button>
        </td>
    </tr>
    <tr>
        <td>
            <button>7</button>
        </td>
        <td>
            <button>8</button>
        </td>
        <td>
            <button>9</button>
        </td>
        <td rowspan="2">
            <button>+</button>
        </td>
    </tr>
    <tr>
        <td>
            <button>4</button>
        </td>
        <td>
            <button>5</button>
        </td>
        <td>
            <button>6</button>
        </td>
    </tr>
    <tr>
        <td>
            <button>1</button>
        </td>
        <td>
            <button>2</button>
        </td>
        <td>
            <button>3</button>
        </td>
        <td rowspan="2">
            <button>=</button>
        </td>
    </tr>
    <tr>

```

```
        <td colspan="2">
            <button>0</button>
        </td>
        <td>
            <button>.</button>
        </td>
    </tr>
</table>
</body>
</html>
```

## 十、DOM操作

### 1、创建jquery对象

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/jquery-3.3.1.js"></script>
    <script>

        // 创建jquery对象
        // let $div = $("<div>");
        // $div.css({
        //     "width":"200px",
        //     "height":"200px",
        //     "border":"1px solid black"
        // });

        // $div.html("aaaaaaaaaaaaa");
        // console.log($div);

        let ulStr = `
            <ul>
                <li>111</li>
                <li>112</li>
                <li>113</li>
                <li>114</li>
            </ul>
        `;

        let $ul = $(ulStr);
        $ul.find("li").eq(2).css("color", "red");
        console.log($ul);

    </script>
</head>

<body>
```

```
</body>
</html>
```

## 2、添加

```
<!--原始HTML-->
<a>
  <c></c>
</a>
```

- `$(a).append(b)`

```
<a>
  <c></c>
  <b></b>
</a>
```

- `$(a).prepend(b)`

```
<a>
  <b></b>
  <c></c>
</a>
```

- `$(a).after(b)`

```
<a></a>
<b></b>
```

- `$(a).before(b)`

```
<b></b>
<a></a>
```

## 3、删除

- `$(a).remove()` 删除a以及子标签
- `$(a).empty()` 删除a的子标签，a不删除

## 4、综合案例

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
```

```

<title>Document</title>
<script src="js/jquery-3.3.1.js"></script>
<script>
    // let doAdd = ()=> {
    //     // 取用户名
    //     let name = $("#username").val();
    //     // 取性别 找到radio选中的
    //     let sex = $(":radio:checked").val();
    //     // 取年龄
    //     let age = $("#age").val();
    //     // 取简介
    //     let intro = $("#intro").val();
    //     // 取爱好 :checked 数组 => 字符串
    //     let hobbs = [];
    //     $(":checkbox:checked").each(function(){
    //         hobbs.push($(this).val());
    //     });
    //     let hobby = hobbs.join(",");
    //     // 创建5个td
    //     let td1 = $("<td>");
    //     td1.html(name);
    //     let td2 = $("<td>");
    //     td2.html(sex);
    //     let td3 = $("<td>");
    //     td3.html(age);
    //     let td4 = $("<td>");
    //     td4.html(intro);
    //     let td5 = $("<td>");
    //     td5.html(hobby);
    //     // 创建一个a 设置href属性 设置标签内容 绑定点击事件
    //     let a = $("<a>");
    //     a.prop("href", "#");
    //     a.html("删除");
    //     a.click(function(){
    //         $(this).parent().parent().remove();
    //     })
    //     // 创建一个td 将a 放到td中
    //     let td6 = $("<td>");
    //     td6.append(a);
    //     // 创建一个tr, 将6个td放到tr中
    //     let tr = $("<tr>");
    //     tr.append(td1);
    //     tr.append(td2);
    //     tr.append(td3);
    //     tr.append(td4);
    //     tr.append(td5);
    //     tr.append(td6);
    //     // 将tr放到表格中
    //     $("table").append(tr);
    // }

```

```

let doAdd = ()=> {

```

```

// 取用户名
let name = $("#username").val();
// 取性别 找到radio选中的
let sex = ($("#radio:checked").val());
// 取年龄
let age = ($("#age").val());
// 取简介
let intro = ($("#intro").val());
// 取爱好 :checked 数组 => 字符串
let hobbs = [];
$("#checkbox:checked").each(function(){
    hobbs.push($(this).val());
});
let hobby = hobbs.join(",");

let tr = `
    <tr>
        <td>${name}</td>
        <td>${sex}</td>
        <td>${age}</td>
        <td>${intro}</td>
        <td>${hobby}</td>
        <td>
            <a href="#" onclick="doDel(this)">删除</a>
        </td>
    </tr>
`;

$("#table").append($(tr));
}

let doDel = (item)=> {
    $(item).parent().parent().remove();
}
</script>
</head>
<body>
<table cellspacing="0" cellpadding="5" border="1" width="100%">
    <tr>
        <td>姓名</td>
        <td>性别</td>
        <td>年龄</td>
        <td>简介</td>
        <td>爱好</td>
        <td>操作</td>
    </tr>
    <tr>
        <td>张三</td>
        <td>女</td>
        <td>22</td>
        <td>xxxxxxx</td>

```



```

        <td>吃饭，睡觉</td>
        <td>
            <a href="#">删除</a>
        </td>
    </tr>
</table>
<hr>

<input type="text" id="username" />
<input name="sex" type="radio" value="男" checked>男
<input name="sex" type="radio" value="女">女
<select id="age">
    <option value="1">1</option>
    <option value="2">2</option>
    <option value="3">3</option>
    <option value="4">4</option>
    <option value="5">5</option>
</select>
<textarea name="" id="intro" cols="30" rows="10"></textarea>
<input name="hobby" type="checkbox" value="吃饭">吃饭
<input name="hobby" type="checkbox" checked value="睡觉">睡觉
<input name="hobby" type="checkbox" value="写bug">写bug

<input type="button" value="添加" onclick="doAdd()">

</body>
</html>

```

## 十一、jQuery动画效果

需要记住的 `hide()` 和 `show()`

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Document</title>
    <script src="js/jquery-3.3.1.js"></script>
</script>
<style>
    div{
        width: 100px;
        height: 100px;
        border:1px solid black;
        background-color:red;
    }
</style>
</head>
<body>
    <div>

</div>

```

```

<button onclick="show()">show()</button>
<button onclick="hide()">hide()</button>
<button onclick="toggle()">toggle()</button>
<button onclick="slideUp()">slideUp()</button>
<button onclick="slideDown()">slideDown()</button>
<button onclick="slideToggle()">slideToggle()</button>
<button onclick="fadeIn()">fadeIn()</button>
<button onclick="fadeOut()">fadeOut()</button>
<button onclick="fadeToggle()">fadeToggle()</button>
<button onclick="doAnimate()">animate()</button>
<script>

```

```

    let div = $("div");

```

```

    let show = ()=>{
        div.show(3000);
    }
    let hide = ()=>{
        div.hide(3000);
    }
    let toggle = ()=>{
        div.toggle(3000);
    }
    let slideUp = ()=>{
        div.slideUp(3000);
    }
    let slideDown = ()=>{
        div.slideDown(3000);
    }
    let slideToggle = ()=>{
        div.slideToggle(3000);
    }
    let fadeIn = ()=>{
        div.fadeIn(3000);
    }
    let fadeOut = ()=>{
        div.fadeOut(3000);
    }
    let fadeToggle = ()=>{
        div.fadeToggle(3000);
    }

    let doAnimate = () => {
        div.animate({
            "borderRadius" : "50%",
            "width":"200px",
            "height":"200px",
            // "backgroundColor":"green"
            "opacity":'0.5'
        }, 3000);
    }

```

```
</script>  
</body>  
</html>
```