

一、Jquery 介绍

官方网站: <http://www.jquery.com>。

What is jQuery?

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: 它是目前市场上最流行的 js 框架。它可以帮助开发人员以最小的 js 代码来实现更加丰富的功能, 更简单的事件, 动画效果等等。除此以外, 它还提供可以解决不同浏览器的 JS 兼容性问题。

二、第一个 HelloWorld 例子

第一步: 在 html 页面引入 jQuery 的两个 js 文件;

-- jQuery 的版本说明:

- 1) 如果使用 jQuery.1.9.0 之前的版本, 只需要下载一个 jquery-xxx.js 文件即可;
- 2) 如果使用 jQuery.1.9.0 之后的版本, 就需要下载两个 js 文件:

 jquery-3.3.1.min.js	2018/1/27 8:12	JScript Script 文件	85 KB
 jquery-migrate-1.4.1.min.js	2018/1/27 8:15	JScript Script 文件	10 KB

第二步: 创建 JQuery 对象。

第三步: 调用 JQuery 对象的方法操作 HTML。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!--导入JQuery的两个文件-->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
      //页面加载完执行指定操作
      $(document).ready(function(){
        alert("文加载完...");
      });
    </script>
  </head>
</html>
```

```
        </script>
    </head>
    <body>

    </body>
</html>
```

三、创建 JQuery 对象

3.1 JQuery 对象与 DOM 对象的关系

1) JQuery 对象是通过 JQuery 包装 DOM 对象后产生的对象。其语法格式：
\$(DOM 对象)或 jQuery(DOM 对象)

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入JQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                var nameNode = document.getElementById("userName");
                alert($(nameNode));
            });
        </script>
    </head>
    <body>
        <input type="text" id="userName" value="小宝"/>
    </body>
</html>
```

注意：\$符号只是 jQuery 的一个别名。它的作用是把一个

2) JQuery 对象不能调用 dom 对象的方法，同样地，dom 对象也不能够调用 jquery 对象的方法；

```
$(document).ready(function() {
    var nodeName = document.getElementById("userName");
    //这里不能够访问dom对象的value属性，所以输出undefined
    //alert($(nodeName).value);
    alert($(nodeName).val());
});
```

3) jQuery 对象可以转化成 dom 对象;

\$("#selector").get(i): 根据索引号, 从 jQuery 对象中取指定的 dom 元素;

\$("#selector").get() :返回一个包含所有 dom 元素的数组;

```
$(document).ready(function() {
    var nodeName = $(document.getElementById("userName"));
    //把jQuery对象转化为DOM对象的四种方式
    //alert(nameNode.get(0));
    //alert(nameNode.get()[0]);
    //alert(nameNode[0]);
    alert(nameNode.toArray()[0]);
});
```

3.2 JQuery 选择器

3.2.1 基本选择器

\$("# ID "): ID 选择器;

\$("#标签名"): 标签选择器;

\$(".CLASS "): class 选择器;

\$("#selector1, selector2, ... "): 组合选择器;

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                //ID选择器
                //alert($("#userName").val());
```

```

        //标签名选择器
        //alert($("#input").val());
        //类选择器
        var nodes = $("#userName, #introduce");
        for (var i = 0; i < nodes.length; i++) {
            alert(nodes[i]);
        }
    });
</script>
</head>
<body>
    <input type="text" value="大宝" class="input"/>
    <div>
        <input type="text" id="userName" value="小宝" class="input"/>
        <textarea id="introduce"></textarea>
    </div>
</body>
</html>

```

`$("#selector").size()`：获取 jQuery 对象中查询到得 dom 元素的个数。

`$("#selector").length`：获取 jQuery 对象中查询到得 dom 元素的个数。

`$("#selector").each()`：迭代 jQuery 对象查询到得所有的 dom 元素。

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            //页面加载完执行
            $(function() {
                /*//获取input元素的个数
                alert($("#input").size());
                //获取input元素的个数
                alert($("#input").length); */

                /*var inputNodes = $("#input");
                for (var i = 0; i < inputNodes.size(); i++) {
                    //获取第i个元素的值
                    //alert(inputNodes[i].value);
                    alert(inputNodes.get(i).value);
                }
            });
        </script>
    </head>
    <body>
        <input type="text" value="大宝" class="input"/>
        <div>
            <input type="text" id="userName" value="小宝" class="input"/>
            <textarea id="introduce"></textarea>
        </div>
    </body>
</html>

```

```

        }*/

        /*
        * each: 迭代jQuery对象查询到得所有的dom元素。
        * i: 代表第i个元素，从0开始计算
        * item: 代表当前dom元素对象
        */
        $("input").each(function(i, item) {
            //输出元素的下标
            //alert(i);
            //输出元素的值
            alert(item.value);
        });
    });
</script>
</head>
<body>
    <input type="text" value="aa"/><br />
    <input type="text" value="bb"/><br />
    <input type="text" value="cc"/><br />
</body>
</html>

```

3.2.2 关系选择器

`$("selector1 selector2")`: 查询父元素里面的所有后代子元素;

`$("selector1 > selector2")`: 查询父元素的直接子元素;

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!-- 导入jQuery的两个文件 -->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                //查找body元素下的所有input元素
                //alert($("body input").length);
                //查找body元素下的直接input子元素
            });
        </script>
    </head>
    <body>
        <div>
            <input type="text" value="aa"/>
            <input type="text" value="bb"/>
            <input type="text" value="cc"/>
        </div>
    </body>
</html>

```

```
        alert($("#body>input").length);
    });
</script>
</head>
<body>
    <input type="text" value="大宝" class="input"/>
    <div>
        <input type="text" id="userName" value="小宝" class="input"/>
        <textarea id="introduce"></textarea>
    </div>
</body>
</html>
```

3.2.3 条件选择器

\$("#selector:first"): 获取第一个元素;

\$("#selector:last"): 获取最后一个元素;

\$("#selector:eq(i)": 根据索引获取元素;

\$("#selector:checked"): 查询有 checked 属性的元素;

\$("#selector:selected"): 查询有 selected 属性的元素;

\$("#selector:hidden"); 查询隐藏的。

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                //查询第一个input元素
                //alert($("#input:first").val());
                //查询最后一个input元素
                //alert($("#input:last").val());
                //查询第三个input元素（下标从0开始）
                //alert($("#input:eq(2)").val());
                //查询有已选中的单选框
                //alert($("#input:checked").val());
```

```

        //查询select元素的选中选项
        //alert($("#select>option:selected").val());
        //查询隐藏元素
        alert($("#input:hidden").val());
    });
</script>
</head>
<body>
    <input type="hidden" value="12"/>
    <input type="text" id="userName" value="小宝" class="input"/><br/>
    <input type="password" id="userPass" value="12345"
class="input"/><br/>
    <input type="radio" name="gender" value="male" checked="true"/>男
<input type="radio" name="gender" value="female"/>女<br/><br />
    <select>
        <option value="gz">广州</option>
        <option value="zs" selected>中山</option>
        <option value="zh">珠海</option>
    </select>
    <textarea id="introduce"></textarea>
</body>
</html>

```

3.2.4 属性选择器

\$("selector[属性名=属性值]")
 \$("selector[属性名!=属性值]")
 \$("selector[属性名^=属性值]")
 \$("selector[属性名\$=属性值]")
 \$("selector[属性名*=属性值]")

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!-- 导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {

```

```

        //查询name属性为gender的所有input元素
        //alert($('input[name=gender]').length);
        //查询value属性不是小宝的input元素
        //alert($("input[value!='小宝']").val());
        //查询所有id以user开头的input元素
        //alert($("input[id^='user']").length);
        //查询id以pass结尾的input元素
        //alert($("input[id$='Pass']").val());
        //查询id包含user的input元素
        alert($("input[id*='user']").length);
    });
</script>
</head>
<body>
    <input type="text" id="userName" value="小宝" class="input"/><br/>
    <input type="password" id="userPass" value="12345"
class="input"/><br/>
    <input type="radio" name="gender" value="male" checked="true"/>男
<input type="radio" name="gender" value="female"/>女<br/><br />
</body>
</html>

```

四、JQuery 的基本操作

4.1 操作属性

4.1.1 获取属性值

```

$("selector").attr("属性名");
$("selector").prop("属性名");

```

➤ attr 和 prop 的区别？

对于 HTML 元素本身的固有属性，使用 prop 方法。

对于 HTML 元素我们自己自定义的 DOM 属性，使用 attr 方法。

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入JQuery的两个文件-->

```



```

    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
        $(document).ready(function() {
            //获取元素额外添加的属性
            alert($("#input[type=radio]").attr("checked"));
            //获取元素的固有属性
            alert($("#input[type=radio]").prop("checked"));
        });
    </script>
</head>
<body>
    <input type="radio"/>吃
    <input type="radio"/>喝
    <input type="radio"/>玩
</body>
</html>

```

4.1.2 设置属性值

```

$("#selector").attr("属性名","属性值");
$("#selector").attr({"属性名": "属性值", "属性名": "属性值"});
$("#selector").prop("属性名","属性值");
$("#selector").prop({"属性名": "属性值", "属性名": "属性值"});

```

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                //$("#img").attr("src", "img/3.jpg").attr("title", "这是妖
");
                $("#img").prop("src", "img/3.jpg").prop("title", "这是妖");
            });
        </script>

```

```
</head>
<body>
    <img/>
</body>
</html>
```

4.1.3 删除属性值

`$("selector").removeAttr("属性名");`

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                $("a").removeAttr("href");
            });
        </script>
    </head>
    <body>
        <a href="http://www.baidu.">百度</a>
    </body>
</html>
```

4.1.4 案例：实现购物车的反选功能

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
```

```
<script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
<script>
    function check(cbNode) {
        //注意cbNode是一个DOM对象，而不是一个jQuery对象
        var cbNodes = $("tbody input[type=checkbox]");
        for (var i = 0; i < cbNodes.length; i++) {
            //把DOM对象转换成jQuery对象
            var cbNode = $(cbNodes[i]);
            if (cbNode.prop("checked")) {
                cbNode.prop("checked", false);
            } else {
                cbNode.prop("checked", true)
            }
        }
    }
</script>
</head>
<body>
    <table border="1">
        <thead>
            <tr>
                <th><input type="checkbox"
onclick="check(this)"/></th>
                <th>商品名称</th>
                <th>价格</th>
            </tr>
        </thead>
        <tbody>
            <tr>
                <td><input type="checkbox"/></td>
                <td>电视机</td>
                <td>3000</td>
            </tr>
            <tr>
                <td><input type="checkbox"/></td>
                <td>电视机</td>
                <td>3000</td>
            </tr>
            <tr>
                <td><input type="checkbox"/></td>
                <td>电视机</td>
                <td>3000</td>
            </tr>
        </tbody>
    </table>
</body>
</html>
```

```
        </tbody>
    </table>
</body>
</html>
```

4.2 操作文本

4.2.1 获取文本

`$("#selector").html()`: 获取元素的 HTML 内容。

`$("#selector").text(...)`: 获取元素的文本内容。

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!--导入jQuery的两个文件-->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <style>
      .red {
        font-size: 24px;
        color: red;
      }

      .green {
        font-size: 24px;
        color: green;
      }

      .blue {
        font-size: 24px;
        color: blue;
      }
    </style>
    <script>
      $(document).ready(function() {
        //获取body的内容，包含HTML标签
        //alert($("#document.body).html());
```

```
        //获取body的内容，不包含HTML标签
        alert($(document.body).text());
    });
</script>
</head>
<body>
    <h1>法拉第</h2>
    宝马
</body>
</html>
```

4.2.2 设置文本

```
$("#selector").html("html 格式的字符串");
$("#selector").text("普通文本格式的字符串");
```

4.2.3 删除文本

```
$("#selector").html("")
$("#selector").text("")
```

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                //设置span标签的内容
                //$("#span").html("<h1>白切鸭</h1>");
                //$("#span").text("<h1>白切狗</h1>");
                //删除span标签的内容
                //$("#span").html("");
                $("#span").text("");
            });
        </script>
    </head>
    <body>
```

```
<span>白切鸡</span>
</body>
</html>
```

4.3 操作 value

4.3.1 设置 value

```
$("#selector").val("值");
$("#selector").attr("value", "值");
$("#selector").prop("value", "值");
```

4.3.2 获取 value

```
$("#selector").val();
$("#selector").attr("value");
$("#selector").prop("value");
```

4.3.3 删除 value

```
$("#selector").val("");
$("#selector").attr("value", "");
$("#selector").prop("value", "");
$("#selector").removeAttr("value");
```

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!-- 导入jQuery的两个文件 -->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
      $(document).ready(function() {
        //alert($("#input[type=text]").val());
        //$("#input[type=text]").val("铁蛋");
        $("#input[type=text]").val("");
      });
    </script>
  </head>
</html>
```

```
});  
</script>  
</head>  
<body>  
    用户名: <input type="text" value="小宝"/>  
</body>  
</html>
```

五、文档操作

5.1 添加元素

`$("#selector").append("html 格式的字符串");` // 往目标元素里面最后面添加子元素

`$("#html 格式的字符串").appendTo("selector");` // 把子元素添加到目标元素里面的最后面

`$("#selector").prepend("html 格式的字符串");` // 往目标元素里面最前面添加子元素

`$("#html 格式的字符串").prependTo("selector");` // 把子元素添加到目标元素里面的最前面

```
<!DOCTYPE html>  
<html>  
    <head>  
        <meta charset="UTF-8">  
        <title></title>  
        <!--导入jQuery的两个文件-->  
        <script type="text/javascript"  
src="js/jquery-3.3.1.min.js"></script>  
        <script type="text/javascript"  
src="js/jquery-migrate-1.4.1.min.js"></script>  
        <script>  
            function addRow() {  
                //往目标元素里面最后面添加子元素  
                //$("#table").append("<tr><td>红楼梦</td></tr>");  
                //把子元素添加到目标元素里面的最后面  
                //$("#<tr><td>红楼梦</td></tr>").appendTo($("#table"));  
                //往目标元素里面最前面添加子元素  
                //$("#table").prepend("<tr><td>红楼梦</td></tr>");  
                //把子元素添加到目标元素里面的最前面  
                //$("#<tr><td>红楼梦</td></tr>").prependTo($("#table"));  
            }  
        </script>
```

```

</head>
<body>
  <table border="1" width="200" cellpadding="0">
    <tr>
      <td>三国演义</td>
    </tr>
    <tr>
      <td>水浒传</td>
    </tr>
    <tr>
      <td>西游记</td>
    </tr>
  </table>
  <input type="button" value="添加行" onclick="addRow()">
</body>
</html>

```

\$("#selector").before("html 格式的字符串"); // 往目标元素外面最前面添加相邻的元素

\$("#selector").after("html 格式的字符串"); // 往目标元素外面最后面添加相邻的元素

```

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!--导入jQuery的两个文件-->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
      function addBefore() {
        $("div").before("java基础".fontcolor("red"));
      }

      function addAfter() {
        $("div").after("数据库技术".fontcolor("red"));
      }
    </script>
  </head>
  <body>
    <div>javaweb入门</div><br/>
    <input type="button" value="在前面添加" onclick="addBefore()"/>

```



```
<input type="button" value="在后面添加" onclick="addAfter()"/>
</body>
</html>
```

5.2 插入元素

`$("#html 格式的字符串").insertBefore("selector");` // 把元素添加到目标元素外面的最前面

`$("#html 格式的字符串").insertAfter("selector");` // 把元素添加到目标元素外面的最后面

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!--导入jQuery的两个文件-->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
      function addBook() {
        var bookNode = $("#<tr><td>红楼梦</td><td>50</td></tr>");
        //往目标元素外面最前面添加相邻的元素
        //bookNode.insertBefore($("#last"));
        //往目标元素外面最后面添加相邻的元素
        bookNode.insertAfter($("#last"));
      }
    </script>
  </head>
  <body>
    <table border="1" width="200" cellspacing="0">
      <tr>
        <th>书名</th>
        <th>价格</th>
      </tr>
      <tr>
        <td>三国演义</td>
        <td>50</td>
      </tr>
      <tr>
        <td>水浒传</td>
```

```

        <td>70</td>
    </tr>
    <tr id="last">
        <td>西游记</td>
        <td>90</td>
    </tr>
</table>
<input type="button" value="添加图书" onclick="addBook()"/>
</body>
</html>

```

5.3 删除元素

`$("#selector").empty()`：删除所有的子元素。

`$("#selector").remove()`：删除元素。

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            function deleteBook() {
                //删除所有的子元素
                //$("table").empty();
                //删除指定元素
                $("tr").remove("#last");
            }
        </script>
    </head>
    <body>
        <table border="1" width="200" cellspacing="0">
            <tr>
                <th>书名</th>
                <th>价格</th>
            </tr>
            <tr>

```

```

        <td>三国演义</td>
        <td>50</td>
    </tr>
    <tr>
        <td>水浒传</td>
        <td>70</td>
    </tr>
    <tr id="last">
        <td>西游记</td>
        <td>90</td>
    </tr>
</table>
<input type="button" value="删除图书" onclick="deleteBook()"/>
</body>
</html>

```

六、筛选

jQuery 可以通过以下方法对选择器进行过滤。

- eq(i): 根据索引取。
- first(): 取第一个
- last(): 取最后一个
- prev(): 获取上一个同辈的元素；
- next(): 获取下一个同辈的原素；
- parent(): 获取父元素；
- children(): 获取所有子元素的集合；

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                /*//查找所有的option对象
                var options = $("select").find("option");

```

```
        for (var i = 0; i < options.length; i++) {
            alert(options[i].text);
        }

        //查找select对象的所有子元素
        var options = $("select").children();
        for (var i = 0; i < options.length; i++) {
            alert(options[i].text);
        }

        //查找选中状态的option
        alert($("select>option").filter(":selected").text());

        //查找没有选中的option
        var options = $("select").find("option").not(":selected");
        for (var i = 0; i < options.length; i++) {
            alert(options[i].text);
        }
    });
</script>
</head>
<body>
    <select>
        <option selected>广东</option>
        <option>湖南</option>
        <option>广西</option>
    </select>
</body>
</html>
```

七、事件

7.1 绑定事件

方式一：通过对象的 on 方法绑定事件

➤ 语法格式：

`$(“selector”).on(“事件名”, 参数, 回调函数);`

第一个参数：事件名把前面的 on 去掉，多个事件名中间用空格分隔开；

第二个参数：传入事件监听函数中的数据 json 格式的 js 对象；

第三个参数：事件处理函数；

```
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!--导入jQuery的两个文件-->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
      $(document).ready(function() {
        $("button").on("click", {name:'张三'}, function(event) {
          alert(event.data.name);
        });
      });
    </script>
  </head>
  <body>
    <button>click me! </button>
  </body>
</html>
```

➤ 事件对象的属性和方法:

event.data: 获取数据(json)

event.currentTarget : 当前的 dom 元素

event.target: 当前的 dom 元素

event.type: 获取事件的类型

event.preventDefault(): 取消事件的默认行为.

event.stopPropagation(): 阻止事件传播.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title></title>
    <!--导入jQuery的两个文件-->
    <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
    <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
    <script>
      $(document).ready(function() {
        /*$("button").on("click", {name:'张三'}, function(event) {
          //获取事件参数
          alert(event.data.name);
        });
      });
    </script>
```

```

        //获取当前事件源对象
        alert(event.currentTarget.innerHTML);
        alert(event.target.innerHTML);
        alert(this.innerHTML);
        //获取事件类型
        alert(event.type);
    });*/

    $("div").on("click", function(event) {
        alert("点击...");
    });

    $("a").on("click", function(event) {
        //取消事件的默认行为
        event.preventDefault();
        //取消事件的传递
        event.stopPropagation();
        alert("呵呵");
    });
});
</script>
</head>
<body>
    <div>
        <!--<button>click me! </button>-->
        <a href="http://www.entor.com.cn">易唐科技</a>
    </div>
</body>
</html>

```

➤ 绑定多个事件:

```

<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                $("a").on({

```

```
        mouseover: function(event) {
            alert("鼠标移入...");
        },
        mouseout: function(event) {
            alert("鼠标移出...");
        }
    });
});
</script>
</head>
<body>
    <a href="http://www.entor.com.cn">易唐科技</a>
</body>
</html>
```

方式二：通过对象的 **one** 方法绑定事件

➤ 语法格式：

`$("selector").one("事件名", 参数, 回调函数);`

第一个参数：事件名把前面的 `on` 去掉，多个事件名中间用空格分隔开。

第二个参数：传入事件监听函数中的数据 `json` 格式的 `js` 对象。

第三个参数：事件处理函数。

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(document).ready(function() {
                /*$("a").one("mouseover", function(event) {
                    alert("鼠标移入...");
                });*/

                $("a").one({
                    mouseover: function(event) {
                        alert("鼠标移入...");
                    },
```

```
        mouseout: function(event) {
            alert("鼠标移出...");
        }
    });
});
</script>
</head>
<body>
    <a href="http://www.entor.com.cn">易唐科技</a>
</body>
</html>
```

注意：使用 one 方法绑定的事件只会被触发一次。

方式三：使用特殊方法绑定事件

jQuery 为每一个事件都提供了 onXxx 方法，Xxx 代表事件名。例如：

onblur --> blur()

onclick --> click()

onmouseover --> mouveover()

```
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <!--导入jQuery的两个文件-->
        <script type="text/javascript"
src="js/jquery-3.3.1.min.js"></script>
        <script type="text/javascript"
src="js/jquery-migrate-1.4.1.min.js"></script>
        <script>
            $(function() {
                $("button").click(function() {
                    alert("click me!");
                });

                $("button").mouseover(function() {
                    alert("鼠标移入..");
                });
            });
        </script>
    </head>
    <body>
```



```
<button>click me</button>
</body>
</html>
```

7.2 移除事件

jQuery 对象提供了 `unbind` 函数，该函数可以把对象上绑定的事件移除。

命令格式：

```
$(“selector”).unbind(“事件名”)
```

注意：如果没有指定事件名，那么就会把该元素的所有事件移除。

八、常用插件

jquery 官方提供了一些插件，方便用户在项目中使用 JQuery 开发页面功能。

官方插件库：<https://plugins.jquery.com/>

8.1 datepicker 插件

Datepicker 插件可以实现在输入框中显示日期选择器。

Datepicker 插件的使用步骤：

第一步：在 html 页面中导入相关文件。

```
<script type="text/javascript" src="js/jquery-3.3.1.min.js" ></script>
<script type="text/javascript" src="js/jquery-migrate-1.4.1.min.js" ></script>
<link rel="stylesheet" href="css/jquery-ui.min.css">
<script src="js/jquery-ui.min.js"></script>
```

第二步：初始化日期插件。

```
<body>
  <input type="text" id="datepicker" readonly="readonly"/>
  <script>
    $(function() {
      $("#datepicker").datepicker();
    });
  </script>
</body>
```

第三步：测试，运行效果如下图所示：

07/03/2019

July 2019

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

第四步：设置日期格式。

```
$(function() {  
    $("#datepicker").datepicker({dateFormat: "yy年mm月dd日"});  
});
```

格式字符代表的含义：

- dd-每月的第几天（两位数字）
- mm - 月份（两位数字）
- yy - 年份（四位数字）

运行效果：

2019年07月10日

July 2019

Su	Mo	Tu	We	Th	Fr	Sa
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

8.2 validation 插件

Validation 插件可以实现表单的验证。

8.2.1 Validation 插件的使用步骤

第一步：在 html 页面中引入相关文件。

```
<script type="text/javascript" src="js/jquery-3.3.1.min.js" ></script>
<script type="text/javascript" src="js/jquery-migrate-1.4.1.min.js" ></script>
<script type="text/javascript" src="js/jquery.validate.min.js" ></script>
```

第二步：初始化插件。

```
<body>
  <form id="registForm">
    用户名: <input type="text" id="userName"/><br />
    密码: <input type="text" id="userPass" /><br />
    手机号码: <input type="text" id="phone" /><br />
    <input type="submit" value="提交"/>
  </form>
  <script>
    $(function() {
      $("#registForm").validate();
    });
  </script>
</body>
```

第三步：在标签上添加校验规则。

```
<form id="registForm">
  用户名: <input type="text" name="userName" required/><br />
  密码: <input type="text" name="userPass" minlength="5" maxlength="10" required/><br />
  手机号码: <input type="text" name="phone" required/><br />
  电子邮箱: <input type="text" name="email" type="email"/><br />
  个人主页: <input type="text" id="url" type="url"/><br />
  <input type="submit" value="提交"/>
</form>
```

第四步：测试，效果如下：

用户名 : This field is required.
密码 : This field is required.
手机号码 : This field is required.
电子邮箱 : 11 Please enter a valid email address.
个人主页 : 11 Please enter a valid URL.

第五步：在 html 页面中引入中文插件。

```
<script type="text/javascript" src="js/jquery-3.3.1.min.js" ></script>
<script type="text/javascript" src="js/jquery-migrate-1.4.1.min.js" ></script>
<script type="text/javascript" src="js/jquery.validate.min.js" ></script>
<script type="text/javascript" src="js/messages_cn.js" ></script>
```

重新测试，运行效果如下：

用户名： 必选字段

密码： 必选字段

手机号码： 必选字段

电子邮箱： 请输入正确格式的电子邮件

个人主页： 请输入合法的网址

8.2.2 校验规则

required:true 必须输入的字段。

remote:"check.php" 使用 ajax 方法调用 check.php 验证输入值。

email:true 必须输入正确格式的电子邮件。

url:true 必须输入正确格式的网址。

date:true 必须输入正确格式的日期。日期校验 ie6 出错，慎用。

dateISO:true 必须输入正确格式的日期（ISO），例如：2009-06-23，
1998/01/22。只验证格式，不验证有效性。

number:true 必须输入合法的数字（负数，小数）。

digits:true 必须输入整数。

creditcard: 必须输入合法的信用卡号。

equalTo:"#field" 输入值必须和 #field 相同。

accept: 输入拥有合法后缀名的字符串（上传文件的后缀）。

maxlength:5 输入长度最多是 5 的字符串（汉字算一个字符）。

minlength:10 输入长度最小是 10 的字符串（汉字算一个字符）。

`rangelength:[5,10]` 输入长度必须介于 5 和 10 之间的字符串（汉字算一个字符）。

`range:[5,10]` 输入值必须介于 5 和 10 之间。

`max:5` 输入值不能大于 5。

`min:10` 输入值不能小于 10。

8.2.3 修改提示内容

```
$.extend($.validator.messages, {  
  
    required: "这是必填字段",  
  
    remote: "请修正此字段",  
  
    email: "请输入有效的电子邮件地址",  
  
    url: "请输入有效的网址",  
  
    date: "请输入有效的日期",  
  
    dateISO: "请输入有效的日期 (YYYY-MM-DD)",  
  
    number: "请输入有效的数字",  
  
    digits: "只能输入数字",  
  
    creditcard: "请输入有效的信用卡号码",  
  
    equalTo: "你的输入不相同",  
  
    extension: "请输入有效的后缀",  
  
    maxlength: $.validator.format("最多可以输入 {0} 个字符"),  
  
    minlength: $.validator.format("最少要输入 {0} 个字符"),  
  
    rangelength: $.validator.format("请输入长度在 {0} 到 {1} 之间的字符串"),
```

```
range: $.validator.format("请输入范围在 {0} 到 {1} 之间的数值"),  
  
max: $.validator.format("请输入不大于 {0} 的数值"),  
  
min: $.validator.format("请输入不小于 {0} 的数值")  
  
});
```

8.2.4 将校验卸载 js 中

```
<!DOCTYPE html>  
<html>  
  <head>  
    <meta charset="UTF-8">  
    <title></title>  
    <script type="text/javascript" src="js/jquery-3.3.1.min.js" ></script>  
    <script type="text/javascript" src="js/jquery-migrate-1.4.1.min.js" ></script>  
    <script type="text/javascript" src="js/jquery.validate.min.js" ></script>  
    <script type="text/javascript" src="js/messages_cn.js" ></script>  
  </head>  
  <body>  
    <form id="registForm">  
      用户名: <input type="text" name="userName" /><br />  
      密码: <input type="text" name="userPass" /><br />  
      手机号码: <input type="text" name="phone" /><br />  
      电子邮箱: <input type="text" name="email" /><br />  
      个人主页: <input type="text" name="url" /><br />  
      <input type="submit" value="提交" />  
    </form>  
    <script>  
      $(function() {  
        $("#registForm").validate({  
          rules: {  
            userName: "required",  
            userPass: {  
              required: true,  
              minlength: 5,  
              maxlength: 10  
            },  
            phone: "required",  
            email: {
```

```
        email: true
    },
    url: {
        url: true
    }
},
messages: {
    userName: "请输入用户名",
    userPass: {
        required: "请输入密码",
        minlength: "密码长度不少于 5 位",
        maxlength: "密码长度不大于 10 位"
    },
    phone: "请输入手机号码",
    email: "请输入合法的邮箱地址",
    url: "请输入合法的网址"
}
});
</script>
</body>
</html>
```

运行效果如下图所示：

用户名：	<input type="text"/>	请输入用户名
密码：	<input type="password" value="1"/>	密码长度不少于5位
手机号码：	<input type="text"/>	请输入手机号码
电子邮箱：	<input type="text" value="1"/>	请输入合法的邮箱地址
个人主页：	<input type="text" value="1"/>	请输入合法的网址
<input type="button" value="提交"/>		