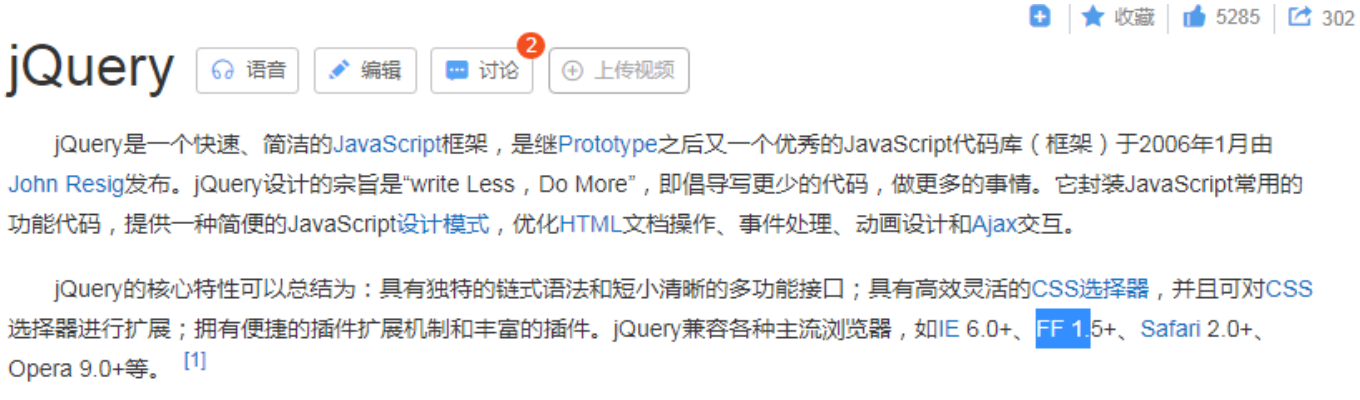
第五章：jQuery

第1节：理解

1. jQuery是js的一个轻量级框架
2. jQuery就是将js常用的功能 封装成了函数(函数库)
3. jQuery写少量的代码 实现更多的功能



第2节：使用

1. 下载函数库 https://jquery.com/ jquery.js文件
2. 将js文件引入项目中
3. 引入函数库到页面

```
<!-- 引入jq函数库 -->
<script src="js/jquery-3.5.1.js"></script>
```

4. 编写jq代码

第3节：核心语法

API文档：https://jquery.cuishifeng.cn/

1. $() jquery的核心函数 根据任意选择器获取对应的标签
2. 原生js的window.onload = function (){} 封装成了 $(function(){})
   ① js中的window.onload只能出现一次 否则会覆盖
   ② 封装后的$(function(){}) 可以出现多次 并且依次执行

3. 原生js对象和jquery对象
   ① 原生js对象只能调用自己的函数和属性 jquery只能调用自己的函数和属性
   ② 原生js对象和jquery对象的相互转换
      1. js对象转换成jquery对象 $(原生的js对象) 返回一个jquery对象
      2. jquery对象转换成js对象
         jquery对象[0] 或者 jquery对象.get(0)

   特点：
      jquery本质上是 一组js对象

```
    <script>
        $(function(){

            // 原生js获取对象  整个标签元素
            var sp = document.getElementById("sp");
            // console.log(sp.innerHTML);

            // jquery获取对象
            console.log( $("#sp").html())
            // console.log( $("span"))

            // 原生的js对象和jquery对象的转换
            // 1. jquery对象转换成js对象
            var s = $("#sp")[0];
            var ss = $("#sp").get(0);
            // console.log(ss.innerHTML)

            // 2. js对象转换成jquery对象
            var sss = $(sp);
            console.log(sss.html());
        })
    </script>
</head>
<body>
    <span id="sp">der雪</span>
    <span></span>
</body>
```

第4节：常见的函数

4.1 选择器

定位到具体的标签 在css选择器的基础上 丰富了一系列的选择器 更加方便简洁的获取标签

分类：

1. 基本选择器

```
    <script>
        $(function(){
            // 1. #id选择器
            $("#d1").css("color","red");

            // 2. .class选择器
            $(".sp").css("color","green");

            // 3. element选择器
            $("p").css("color","yellow");

            // 4. 组合选择器
            $("#d1,.sp,p").css("background-color","blue");
        })
    </script>
</head>
<body>
    <div id="d1">
        这是der祥
    </div>
    <span class="sp">
        这是der雪
    </span>
    <p>
        这是der展
    </p>
</body>
```

2. 层级选择器

```
// 层级选择器
// 1. a b 获取a标签的所有后代
$("ul li").css("color","pink");
console.log($("ul li").length);

// 2. a > b 获取a标签的第一代子标签b
console.log($("ul>li").length);
console.log($(".d2>div").length);

// 3. a + b 获取a标签后面相邻的兄弟标签
// $(".d2+div").css("color","red");

// 4. a ~ b 获取a标签的后面的所有兄弟标签
$(".d2 ~ div").css("color","green");
```

3. 基本筛选器

```
// 基本选择器
// 1. :first 匹配第一个元素
$("ol li:first").css("color","pink");

// 2. :last  匹配最后一个元素
$("ol li:last").css("color","blue");

// 3. :not(selector) 除了匹配的选择器的元素
$("ol li:not(.xx)").css("background-color","red");

// 4. :even 匹配偶数索引的元素
$(".ul li:even").css("background-color","blue");

// 5. :odd
$(".ul li:odd").css("background-color","red");

$("table tr:even:gt(0)").css("background-color","pink");
$("table tr:odd").css("background-color","green");

// 6. :eq(index) 匹配具体索引值的元素
$(".ul li:eq(2)").css("background-color","yellow");

// 7. :gt(index) 匹配索引值大于index的元素
$(".ul li:gt(2)").css("background-color","white");

// 8. :lt(index) 匹配索引值小于index的元素
$(".ul li:lt(2)").css("background-color","#000000");
```

4. 属性选择器

```
// 属性选择器
$("span[class]").css("color","pink"); // 匹配包含class属性的元素
$("span[class=xx1]").css("color","red"); // 匹配class属性值为xx1的元素
$("span[class!=xx2]").css("color","green"); // 匹配class属性值不为xx2的元素
$("span[class^=xx]").css("color","blue"); // 匹配属性值以xx开头
$("span[class$=3]").css("color","pink"); // 匹配属性值以3结尾
$("span[class*=a]").css("color","#0000002"); // 匹配属性值包含a的元素
```

5. 其他选择器

子元素
:nth-child
:first-child
:last-child
:only-child
表单
:input
:text
:password
:radio
:checkbox
:submit
:image
:reset
:button
:file
:hidden
表单对象属性
:enabled
:disabled
:checked
:selected

4.2 操作函数

1. 属性操作 attr 和 prop
    1. attr(参数) 获取元素的属性值
    2. attr(参数1,参数2) 设置元素属性值
    3. attr({}) 设置元素多个属性值

```
    <script src="js/jquery-3.5.1.js" ></script>
    <script>
        $(function(){
            // 获取属性值
            var v = $("#d1").attr("id");
            console.log(v);

            // 设置属性值
            $("#d1").attr("xx","xxxxx");

            // 设置多个属性值
            $("#d1").attr({"x":"haha","xxx":"heihei","xxxx":"hehe"});
        })
    </script>
</head>
<body>
    <div id="d1"></div>
    <img />
    <button onclick="showImg()">点击按钮显示图片</button>
    <script>
        function showImg(){
            $("img").attr({"src":"img/yy.jpg","title":"岩岩老师","width":"100px","height":"10
        }
    </script>
</body>
</html>
```

2. 属性操作
    1. prop(参数) 获取元素的属性值
    2. prop(参数1,参数2) 设置元素属性值
    3. prop({}) 设置元素多个属性值

attr和prop区别:
    1. 设置属性值 prop不写单位
    2. prop获取checked属性 返回的是true或者false 而 attr返回的是checked或者undefined, 因此prop方便做判断处理

3. 内容操作
    html(参数): 如果无参 就是获取双边标签的内容 如果有参 设置标签内容
    text(参数): 如果无参 就是获取双边标签的文本内容 如果有参 设置标签内容
    val(参数): 如果无参 就是获取input的内容 如果有参 设置标签内容

4. 样式操作
    css(参数) 获取标签元素的样式值
    css(样式名,样式值) 设置标签的一个样式
    css({样式名:样式值,样式名:样式值....}) 设置表的多个样式

```
    <script src="js/jquery-3.5.1.js"></script>
    <script>
        $(function(){

            // 1. 获取样式值
            console.log($(".d1").css("width"));

            // 2. 设置样式值
            $(".d1").click(function(){
                // $(this).css("background-color","green");
                $(this).css({"background-color":"green","width":"200px","height":"200px"});
            })
        })
    </script>
</head>
<body>
    <div class="d1" style="width: 100px; height: 100px; background-color: blue;">

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

5. 事件

将js的事件名 去除on 封装成以事件名为函数名的函数

如:

```
js事件     -------->          jquery事件
xx.onclick = function(){} -----> xx.click(function(){})
xx.ondblclick = function(){} ----> xx.dblclick(function(){})
xx.onchange = function(){} ----> xx.change(function(){})
......
```

6. 文档处理

① 内部追加

```
// 内部插入
// 1. 将li追加到ul的内部后面
// $("ul").append($(".li1"));

// 2. 将ul追加到li的内部后面
// $("ul").appendTo($(".li1"));

// 3. 将li插入到ul的内部的前面
// $("ul").prepend($(".li1"));

// 4. 将ul插入到li的内部的前面
// $("ul").prependTo($(".li1"));

// 外部插入
//  1. 将 li1插入到 li2的后面
// $(".li2").after($(".li1"));
```

② 外部插入

```
// 外部插入
//  1. 将 li1插入到 li2的后面
// $(".li2").after($(".li1"));


// 2. 将 li1插入到 li2的前面
// $(".li2").before($(".li1"));


// 3. 将li2插入到 li1的前面
// $(".li2").insertBefore($(".li1"));

//  4. 将li2插入到 li1的后面
// $(".li2").insertAfter($(".li1"));
```

3. 包裹

```
// 包裹
// 1. 将li1标签的外面包裹一个b标签
// $("li").wrap("<b></b>");

// 2. 将所有的li放在一起 在外部包裹 i标签
// $("li").wrapAll("<i></i>");

// 3. 将匹配到的li的内部分别包裹一个i标签
// $("li").wrapInner("<i></i>");
```

4. 替换 删除 复制

```
// 替换
// 1. 将匹配到的li标签替换成了 p标签
// $("li").replaceWith("<p></p>");
// $("<p></p>").replaceAll($("li"));

// 删除
// 1. 清除匹配元素内部的内容
// $("li").empty();

// 2. 删除匹配到的元素
// $("li").remove();

// 复制
var v = $(".li1").clone();
console.log(v)

$("ul").append(v);
```

5. jquery的遍历

格式:

```
$.each(要遍历jquery对象,function(每一个索引, 每一个js对象){})
要遍历jquery对象.each(function(每一个索引, 每一个js对象){})
```

```
<script>
    $(function(){
        // 1. 获取匹配到的所有的li的内容
        // console.log($("ul li").html());
        var lis = $("ul li");
        // 遍历的格式:
        // 1. $.each(jquery对象, function(index, element){})
        // 2. jquery.each(function(index, element){})
        $.each(lis, function(index, el){

            console.log(index+":::"+el.innerHTML);

        })
        console.log("------------------------")
        lis.each(function(index, el){
            console.log(index+":::"+el.innerHTML);
        });
    })
</script>
ead>
dy>
<ul>
    <li>红警</li>
    <li>魔兽世界</li>
    <li>魔兽争霸</li>
    <li class="li2">跑跑卡丁车</li>
    <li>开心消消乐</li>
    <li>王者荣耀</li>
    <li>吃鸡</li>
</ul>
ody>
```

第5节：jquery插件

表单校验插件：

在学习javascript时候，我们手动的完成过表单数据的校验，这项功能在实际开发中也是常见的，属于通用功能，但是单纯的通过javascript进行校验，如果选项过多，那么还是有些力不从心。实际在开发中我们都是使用第三方工具，本案例中我们将使用jQuery插件validation进行表单的校验。

validation插件的使用步骤：

1、下载validation工具。

下载地址：https://github.com/jquery-validation/jquery-validation/releases/download/1.17.0/jquery-validation-1.17.0.zip

2、导入工具jquery-3.4.1.js、jquery.validate.js、messages_zh.js。

3、编写form表单信息并在脚本中给form表单绑定validate验证方法。

4、在form表单元素中逐个指定校验规则。

5、实现注册表单校验案例

序号	规则	描述
1	required:true	必须输入的字段。
2	remote:"check.php"	使用 ajax 方法调用 check.php 验证输入值。
3	email:true	必须输入正确格式的电子邮件。
4	url:true	必须输入正确格式的网址。
5	date:true	必须输入正确格式的日期。日期校验 ie6 出错，慎用。
6	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。
7	number:true	必须输入合法的数字（负数，小数）。
8	digits:true	必须输入整数。
9	creditcard:	必须输入合法的信用卡号。
10	equalTo:"#field"	输入值必须和 #field 相同。
11	accept:	输入拥有合法后缀名的字符串（上传文件的后缀）。
12	maxlength:5	输入长度最多是 5 的字符串（汉字算一个字符）。
13	minlength:10	输入长度最小是 10 的字符串（汉字算一个字符）。
14	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）。
15	range:[5,10]	输入值必须介于 5 和 10 之间。
16	max:5	输入值不能大于 5。
17	min:10	输入值不能小于 10。

jquery插件网址：www.jq22.com
前端学习网站：www.w3school.com.cn　　www.runoob.com