一、jQuery简介

1. 介绍

jQuery是JS的工具库，对原生JS中的DOM操作、事件处理、包括数据处理和Ajax技术等进行封装,提供更完善,更便捷的方法。

2. 使用

1）引入

先引入jquery文件，才能使用jquery语法

2）工厂函数 - $()

"$()"函数用于获取元素节点，创建元素节点或将原生JS对象转换为jquery对象,返回 jQuery 对象。jQuery 对象实际是一个类数组对象，包含了一系列 jQuery 操作的方法。 例 :

```
//$()获取元素节点,需传入字符串的选择器
$("h1")
$("#d1")
$(".c1")
$("body,h1,p")
```

3）原生JS对象与jQuery对象

原生JS对象与jQuery对象的属性和方法不能混用。可以根据需要，互相转换 :

1. 原生JS转换jQuery对象 $(原生对象)，返回 jQuery 对象
2. jQuery对象转换原生JS对象
    - 方法一 : 根据下标取元素,取出即为原生对象 var div = $("div")[0];

- 方法二 : 使用jQuery的get(index)取原生对象 var div2 = $("div").get(0);

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <!--引入jquery文件-->
    <script src="jquery-1.11.3.js"></script>
    <script>
        window.onload = function (){
            //使用jquery获取元素节点
            //$("选择器")
            console.log($("h1"));
            $("h1").html("haha");
            console.log($("h1")[0]);//获取原生JS对象
            //原生对象调用原生的方法
            $("h1")[0].innerHTML = "hehe";
            //eq(index)返回指定下标对应的jq对象
            console.log($("h1").eq(1));
            $("h1").eq(1).html("jq对象调用jq方法");
            console.log(test);
            //原生对象转换jquery对象,$(原生)封装
            $(test).html("包装成jquery对象");
        };
    </script>
</head>
<body>
    <h1>jquery</h1>
    <h1>jquery</h1>
    <h1 id="test">jquery</h1>
    <h1>jquery</h1>
    <h1>jquery</h1>
</body>
</html>
```

4）jQuery获取元素

jQuery通过选择器获取元素，$("选择器") 选择器分类 :

1. 基础选择器

```
标签选择器: $("div")
ID 选择器: $("#d1")
类选择器: $(".c1")
群组选择器: $("body,p,h1")
```

2. 层级选择器

```
后代选择器:  $("div .c1")
子代选择器:  $("div>span")
相邻兄弟选择器:  $("h1+p")  匹配选择器1后的第一个兄弟元素,同时要求兄弟元素满足选择器2
通用兄弟选择器:  $("h1~h2") 匹配选择器1后所有满足选择器2的兄弟元素
```

3. 过滤选择器 需要结合其他选择器使用。

```
:first
  匹配第一个元素 例:$("p:first")
:last
  匹配最后一个元素 例:$("p:last")
:odd
  匹配奇数下标对应的元素
:even
  匹配偶数下标对应的元素
:eq(index)
  匹配指定下标的元素
:lt(index)
  匹配下标小于index的元素
:gt(index)
  匹配下标大于index的元素
:not(选择器)
  否定筛选,除()中选择器外,其他元素
```

4. 属性选择器 属性选择器以[]为标志.

```
1. [attrName]
  匹配包含指定属性的元素
2. [attrName=value]/[attrName="value"]
  匹配属性名=属性值的元素
3. [attrName^=value]
  匹配属性值以指定字符开头的元素
4. [attrName$=value]
  匹配属性值以指定字符结尾的元素
5. [attrName*=value]
  匹配属性值包含指定字符的元素
```

5. 子元素过滤选择器

```
:first-child
   匹配第一个子元素
:last-child
   匹配最后一个子元素
:nth-child(n)
   匹配第n个子元素(n从1开始计数)
```

```
<!doctype html>
<html lang="en">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <!--引入jquery文件-->
    <script src="jquery-1.11.3.js"></script>
    <script>
        window.onload = function (){
            //1.标签选择器
            console.log($("h1").html());
            //2.id选择器
            //css()操作行内样式
            $("#d1").css("color","red");
            //3.class选择器
            $(".c1").css("color","pink");
            //4.群组选择器
            $("body,h1").css("margin","0");
            //5.后代选择器
            $("#box span").css("color","red");
            //6.子代选择器
            $("#box>span").css("background","green");
            //7.相邻兄弟选择器
            $("#box p+span").css("border","1px solid black");
            //8.通用兄弟选择器
            $("#box p~span").css("font-size","32px");
        };
    </script>
</head>
<body>
    <h1 id="d1" class="c1">大旭</h1>
    <h1 class="c1">超哥哥</h1>
    <div id="box">
        <span>div->span</span>
        <p>
            <span>div->p->span</span>
        </p>
        <b>测试</b>
        <span>div->span</span>
        <span>div->span</span>
    </div>
</body>
</html>
```

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
```

```
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <!--引入jquery文件-->
    <script src="jquery-1.11.3.js"></script>
</head>
<body>
    <h1 id="d1">过滤选择器</h1>
    <h1>过滤选择器</h1>
    <h1>过滤选择器</h1>
    <h1 class="c1">过滤选择器</h1>
    <h1>过滤选择器</h1>
    <h1 class="c1">过滤选择器</h1>
    <h1>过滤选择器</h1>
    <h1>过滤选择器</h1>
    <button id="btn1">:first</button>
    <button id="btn2">:last</button>
    <button id="btn3">:odd</button>
    <button id="btn4">:even</button>
    <button id="btn5">:eq()</button>
    <button id="btn6">:lt()</button>
    <button id="btn7">:gt()</button>
    <button id="btn8">:not()</button>
    <script>
        btn8.onclick = function (){
            //匹配下标不为3的所有h1
            //$("h1:not(:eq(3))").css("color","#999");
            $("h1:not(#d1,.c1)").css("color","#ff0");
        };
        btn5.onclick = function (){
            $("h1:eq(3)").css("color","#fa3377");;
        };
        btn6.onclick = function (){
            $("h1:lt(3)").css("color","#33fa77");;
        };
        btn7.onclick = function (){
            $("h1:gt(3)").css("color","#3377fa");
        };

        btn3.onclick = function (){
            $("h1:odd").css("color","blue");;
        };
        btn4.onclick = function (){
            $("h1:even").css("color","orange");;
        };
        btn1.onclick = function (){
            $("h1:first").css("color","red");
        };
        btn2.onclick = function (){
            $("h1:last").css("color","green");
        };
    </script>
</body>
</html>
```

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <!--引入jquery文件-->
    <script src="jquery-1.11.3.js"></script>
</head>
<body>
    <h1>小泽</h1>
    <h1 id="d1">大旭</h1>
    <h1 id="d2">超哥哥</h1>
    <h1 id="cd">老祁</h1>
    <h1 id="adi">铭铭</h1>
    <ul>
        <li>列表项</li>
        <li>列表项</li>
        <li>列表项</li>
        <li>列表项</li>
    </ul>
    <ul>
        <li>列表项</li>
        <li>列表项</li>
        <li>列表项</li>
        <li>列表项</li>
        <a href=""></a>
    </ul>
    <script>
        console.log($("[id]"));
        console.log($("[id='d1']"));
        //匹配id属性值以"d"开头的元素
        console.log($("[id ^= d]"));
        //以"d"结尾
        console.log($("[id $= d]"));
        //id属性值包含"d"
        console.log($("[id *= d]"));

        //子元素过滤选择器
        $("li:first").css("color","red");
        $("li:first-child").css("background","green");
        $("li:last-child").css("background","orange");
        $("li:nth-child(2)").css("text-align","center");
    </script>
</body>
</html>
```

5）操作元素内容

```
html() //设置或读取标签内容,等价于原生innerHTML,可识别标签语法
text() //设置或读取标签内容,等价于innerText,不能识别标签
val()  //设置或读取表单元素的值,等价于原生value属性
```

6）操作标签属性

1. attr("attrName","value") 设置或读取标签属性
2. prop("attrName","value") 设置或读取标签属性 注意 :在设置或读取元素属性时,attr()和prop()基本没有区别;但是在读取或设置表单元素(按钮)的选中状态时,必须用prop()方法,attr()不会监听按钮选中状态的改变,只看标签属性checked是否书写
3. removeAttr("attrName") 移除指定属性

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="jquery-1.11.3.js"></script>
    <script>
        /*
        创建输入框,按钮和div,按钮点击时
        将输入框的值以一级标题的形式显示在div中
        */
        window.onload = function (){
            btn.onclick = function (){
                $("div").text("<h1>"+ $("input").val()+"</h1>")
                //修改输入框的值
                //$("input").val("haha");
            };
            //属性操作
            //jquery链式调用,对同一个jquery对象连缀调用不同方法进行操作
            $("div").attr("id","box").prop("class","c1")
            .html("属性操作").css("color","red");

            getStatus.onclick = function (){
                //获取复选框的选中状态
                console.log($("#savePwd").attr("checked"));
                console.log($("#savePwd").prop("checked"));
                //移除指定属性
                $("#savePwd").removeAttr("checked");
            };
            /*
            原生JS实现:
                1.轮播图实现索引跟随切换,鼠标移入停止,鼠标移出重启定时器,点击
                左右按钮时切换图片
                2.实现全选和取消全选功能
                //附加:
                反选功能
```

```
            */
        };

    </script>
</head>
<body>
    <input type="text">
    <button id="btn">显示</button>
    <div></div>
    <input type="checkbox" id="savePwd" checked>
    <button id="getStatus">获取</button>

</body>
</html>
```

7）操作标签样式

1. 为元素添加id/class属性,对应选择器样式
2. 针对类选择器,提供操作class属性值的方法

```
addClass("className")   //添加指定的类名
removeClass("className")//移除指定的类型,如果参数省略,表示清空class属性值
toggleClass("className")//结合用户行为,实现动态切换类名.如果当前元素存在指定类名,则移除;不存在则添加
```

3. 操作行内样式

```
css("属性名","属性值")  //设置行内样式
css(JSON对象)           //设置一组CSS样式
/*
JSON对象:常用数据传输格式
语法 :
   {
    "width":"200px",
    "height":"200px",
    "color":"red"
   }
 */
```

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="jquery-1.11.3.js"></script>
    <style>
        #d1{
            color:red;
```

```
        }
        .c1{
            background:green;
        }
        .c2{
            text-align:center;
        }
    </style>
    <script>
        window.onload = function ()
        {
            //1.操作id/class属性对应选择器样式
            $("h1").attr("id","d1").attr("class","c1 c2");
            //2.专门来操作属性值,增加移除class属性值//追加
            $("div").addClass("c1").addClass("c2").removeClass("c1");
            //toggleClass()借助用户的行为切换类名,有则删除,无则添加
            demo.onclick = function ()
            {
                $("div").toggleClass("c1 c2");
            }
            //3.操作行内样式
            $("div").css("width","200px");
            $("div").css({
                "height":"200px",
                "border":"1px solid cyan"
            });
            $("div").css("height","300px");//单独操作,其他样式不会受影响

        };
    </script>
</head>
<body>
    <h1>张国荣</h1>
    <div>直接修改</div>
    <button id="demo">切换</button>
</body>
</html>
```

8）根据层级结构获取元素

1. parent() 返回父元素
2. parents('selector') 返回满足选择器的祖先元素
3. children()/children("selector") 返回所有直接子元素/返回满足选择器的直接子元素
4. find("selector") 返回所有的后代元素(包含直接与间接)
5. next()/next("selector") 返回下一个兄弟元素/返回下一个兄弟元素,必须满足选择器
6. prev()/prev("selector") 返回前一个兄弟元素/返回前一个兄弟元素,要求满足选择器
7. siblings()/siblings("selector") 返回所有的兄弟元素/满足选择器的所有兄弟元素

9）元素的创建,添加,删除

1. 创建 使用$("标签语法"), 返回创建好的元素

```
var div = $("<div></div>"); //创建元素
div.html("动态创建").attr("id","d1").css("color","red"); //链式调用，设置内容和属性
var h1 = $("<h1 id='d1'>一级标题</h1>");    //创建的同时设置内容，属性和样式
```

2. 添加至页面 1）作为子元素添加

```
$obj.append(newObj);    //在$obj的末尾添加子元素newObj
$obj.prepend(newObj);   //作为第一个子元素添加至$obj中
```

2）作为兄弟元素添加

```
$obj.after(newObj);     //在$obj的后面添加兄弟元素
$obj.before(newObj);    //在$obj的前面添加兄弟元素
```

3）移除元素

```
$obj.remove();  //移除$obj
```

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="jquery-1.11.3.js"></script>
</head>
<body>
    <div id="box">
        <b>加粗</b>
        <span>参考</span>
        <p>
            <span>
                div->p->span
            </span>
        </p>
        <span>参考</span>
        <b>加粗</b>
    </div>
    <script>
    /*
        //获取父元素
        console.log($("p").parent());

        //获取祖先元素//多层嵌套的情况,满足选择器的父元素
        console.log($("p").parents(),$("p").parents("#box"));

        //获取直接子元素数组
```

```
        console.log($("div").children(),$("div").children("p"));

        //获取满足选择器(参数必须填)所有后代元素
        console.log($("div").find("span"));

        //获取前后相邻兄弟元素next()/prev();//只能找到一个
        console.log($("p").next(),$("p").prev());
        console.log($("p").next("b"),$("p").prev("b"));//结果为0

        //获取所用的兄弟元素
        console.log($("p").siblings());
        console.log($("p").siblings("b"));
    */

        //元素的创建
        var h1 = $("<h1 id='d2'>动态创建</h1>");
        var h2 = $("<h2></h2>");
        h2.html("二级标题").css("color","cyan").attr("id","d3");
        console.log(h1,h2);

        //元素节点的添加
        //作为子元素添加到父元素的头尾位置
        $("#box").append(h1);//末尾
        $("#box").prepend(h2);//开头

        //脱离父元素,作为兄弟元素
        var h1 = $("<h1 id='d2'>动态创建</h1>");
        var h2 = $("<h2></h2>");
        h2.html("二级标题").css("color","cyan").attr("id","d3");
        console.log(h1,h2);
        $("#box").after(h2);
        $("#box").before(h1);

        /*
            创建h3h4元素节点,作为div的头尾子元素添加页面显示,在创建超链接
            h3之前, h4之后
        */
        var h3 = $("<h3 id='h3'>动态创建</h3>");
        var h4 = $("<h4></h4>");
        h4.html("二级标题").css("color","cyan").attr("id","h4");

        $("#box").prepend(h3);
        $("#box").append(h4);

        var a1 = $("<a href=''>a1</a>")
        var a2 = $("<a href=''>a2</a>")
        $("#h3").after(a1);
        $("#h4").before(a2);

        //删除节点
        h3.remove();

    </script>
```

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

10）jQuery事件处理

1. 文档加载完毕
2. 原生JS 方法：window.onload jQuery:

```
//语法一
$(document).ready(function (){
  //文档加载完毕后执行
})
//语法二
$().ready(function (){
  //文档加载完毕后执行
})
//语法三
$(function(){
  //文档加载完毕后执行
})
```

区别： 原生onload事件不能重复书写，会产生覆盖问题；jquery中对事件做了优化,可以重复书写ready方法,依次执行

2. 事件绑定方式 事件名称省略 on 前缀

```
  //on("事件名称", function)
  $("div").on("click",function(){});
  //bind("事件名称",function)
  $("div").bind("click",function(){});
  //事件名作为方法名
  $("div").click(function(){});
```

3. this表示事件的触发对象，在jquery中可以使用，注意转换类型。this为原生对象只能使用原生的属性和方法，可以使用$(this)转换为jquery对象，使用jquery方法。

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="jquery-1.11.3.js"></script>
    <script>
        /*
        //多人开发
        window.onload = function()
```

```
{
    alert("1");
};
if(window.onload)
{
    var fn = window.onload;
    console.log(fn);
    fn();
}
window.onload = function()
{
    alert("2");
};
console.log(window);
*/

//1.jquery中使用ready()方法表示文档加载完毕
/*
$(document).ready(function ()
{
    alert("1");
})
$().ready(function ()
{
    alert("2");
})
$(function ()
{
    alert("3");
})
*/
//2.事件绑定方法
$(function ()
{
    $("h1").on("click",function()
    {
        alert("h1被点击");
    });
    $("h2").bind("click",function()
    {
        alert("h2被点击")
    })
    $("h3").click(function()
    {
        //alert("h3被点击")
    }).mouseover(function()
    {
        console.log("鼠标移入")
    }).mouseout(function()
    {
        console.log("鼠标移除")
    }).css("background","cyan")
})
```

```
    </script>
</head>
<body>
    <h1>标题1</h1>
    <h2>标题2</h2>
    <h3>标题3</h3>
    <ul>
        <li>
            <span>北京</span>
            <ol>
                <li>北京</li>
                <li>上海</li>
                <li>广州</li>
                <li>深圳</li>
            </ol>
        </li>
    </ul>
    <script>
        //下拉菜单的点击传值
        $("ol li").click(function()
        {
            //获取当前被点击的元素的标签内容
            //console.log($(this).html(),this.innerHTML)
            $("span").html($(this).html());
        })
    </script>
</body>
</html>
```

1.轮播图实现索引跟随切换,鼠标移入停止,鼠标移出重启定时器,点击左右按钮时切换图片

```
<!doctype html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport"
          content="width=device-width, user-scalable=no, initial-scale=1.0, maximum-
scale=1.0, minimum-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="jquery-1.11.3.js"></script>
    <script src="banner.js"></script>
    <style>
        /*清除浏览器的默认样式*/
        body,ul,ol{
            margin:0;
            padding:0;
            /*取消列表默认样式（项目符号）*/
            list-style:none;
        }
        body{
             color:#ddd;
        }
```

```
        a{
            text-decoration:none;
            color:#ddd;
        }
        /*外围结构的样式*/
        #nav,#banner{
            width:990px;
            margin:0 auto;
        }
        /*单独设置每个模块的样式*/
        /*1.导航栏*/
        #nav{
            background:green;
            /*解决子元素全部浮动,父元素高度为0*/
            height:30px;
        }
        /*匹配ul中的直接子元素*/
        #nav ul>li{
            float:left;
            margin-left:50px;
            /*设置垂直居中*/
            line-height:30px;、

        }
        #nav ul>li:first-child{
            position:relation;
        }
        #nav ol{
            /*隐藏*/
            display:none;
            position:absolute;
            z-index:10;
        }
        /*子元素过滤选择器,li:first-child匹配
        作为第一个子元素存在的li*/
        #nav ul>li:first-child:hover ol{
            background:red;
            display:block;
        }
        #nav ol li:hover{
            background:orange;
        }
        /*2.轮播图*/
        #banner{
            margin-top:30px;
            height:300px;
            position:relative;
        }
        #banner>img{
            width:990px;
            height:300px;
        }
        #banner ul{
```

```
            width:140px;
            position:absolute;
            bottom:20px;
            /*参照父元素的宽度计算偏移量*/
            left:50%;
            margin-left:-75px;
            /*background:green;*/

        }
        #banner li{
            float:left;
            width:20px;
            height:20px;
            background:gray;
            border-radius:50%;
            margin-left:10px;
        }
        #banner ul>li:first-child{
            margin:0;
            background:red;
        }
        #banner a{
            /*元素脱离之后,手动可以调整尺寸,默认大小由内容决定*/
            position:absolute;
            width:60px;
            height:30px;
            background:orange;
            top:50%;
            margin-top:-15px;
        }
        .prev{
            left:10px;
        }
        .next{
            right:10px;
        }


    </style>
</head>
<body>
    <div id="nav">
        <ul>
            <li>
                北京
                <ol>
                    <li>北京</li>
                    <li>上海</li>
                    <li>广州</li>
                    <li>深圳</li>
                </ol>
            </li>
            <li>订单查询</li>
```

```
            <li>配送范围</li>
        </ul>
    </div>
    <div id="banner">
        <img src="image/index_banner1.jpg" alt="">
        <ul>
            <li></li>
            <li></li>
            <li></li>
            <li></li>
            <li></li>
        </ul>
        <!--阻止超链接的默认的跳转行为href="javascript:void(0)"
        才能添加自定义点击事件-->
        <a href="javascript:void(0)" class="prev">上一张</a>
        <a href="javascript:void(0)" class="next">下一张</a>
    </div>
</body>
</html>
```

```
$(function()
{
    //1.定义数组, 保存图片路径
    var list = ["image/index_banner1.jpg",
    "image/index_banner2.jpg",
    "image/index_banner3.jpg",
    "image/index_banner4.jpg",
    "image/index_banner5.jpg"
    ]
    //2.初始索引
    var index = 0;
    //3.开启定时器
    var timer = setInterval(autoPlay,1000);
    function autoPlay()
    {
        $("#banner li").eq(index).css("background","gray");
        index++;//更新索引
        if(index==list.length)
        {
            index = 0;
        }
        //index=++index==list.length? 0:index;
        $("#banner img").attr("src",list[index]);

        //切换图片索引样式
        $("#banner li").eq(index).css("background","red");
    }

    //鼠标的移入和移除
    $("#banner").mouseover(function()
    {
        clearInterval(timer);
    })
```

```
    $("#banner").mouseout(function()
    {
        timer = setInterval(autoPlay,1000)
    })

    //左右超链接点击切换图片
    $(".prev").click(function()
    {
        $("#banner li").eq(index).css("background","gray");
        index--;
        if(index<0)
        {
            index=list.length-1
        }
        //index = --index<0?list.length-1:index;

        $("#banner img").attr("src",list[index]);

        //切换图片索引样式
        $("#banner li").eq(index).css("background","red");

    })

    $(".next").click(function()
    {
//        $("#banner li").eq(index).css("background","gray");
//        index=++index==list.length? 0:index;
//        $("#banner img").attr("src",list[index]);
//        //切换图片索引样式
//        $("#banner li").eq(index).css("background","red");
        autoPlay();

    })
})
```