

# 多条件筛选的实现

## 表单法

```
<form action="" method="get">
  时代 :
  <label><input type="radio" name="age" value="01" checked>01</label>
  <label><input type="radio" name="age" value="02">02</label>
  <label><input type="radio" name="age" value="03">03</label>
</form>
```

时代： ☒ 01 ☐ 02 ☐ 03

## js中表单数据序列化方式

一共有以下三种：

```
var obj1 = $('#queryForm').serialize();
var obj2 = $('#queryForm').serializeArray();
var obj3 = $('#queryForm').serializeObject();
var obj4 = JSON.stringify(obj3); //通过3转化为json字符串
```

分别对应的是：

obj1: 字符串拼接

obj2: 对象数组,都是name , vlaue

obj3: 对象

obj4: json字符串

## 基础版实现

通过去除 input[type=radio]的默认样式和jQuery动态添加选中项目的label的背景色，实现多条件筛选效果和输出序列化字符串。

时代： 1990 1991 1992

性别： 男 女

age=1992&sex=woman

```
<!DOCTYPE html>
```

```

<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Document</title>
  <script src="https://cdn.bootcss.com/jquery/3.4.1/jquery.min.js"></script>
  <style>

    #form1>div{
      padding: 3px;margin-bottom: 2px;
    }
    #form1 input[type=radio]{
      /*去除浏览器默认样式*/
      -webkit-appearance: none;
      -moz-appearance: none;
      appearance: none;
      display: none;
    }
    #form1 label{
      width: 100px;height: 20px;padding: 3px;
      border-radius: 3px;
      text-align: center;
    }

  </style>
  <script>
    $(function(){
      var chkBgcolor="rgb(222, 140, 69)";
      //显示默认已经选中的项
      $("#form1 input[type=radio]:checked").each(function(){
        $(this).parent("label").css("background-color",chkBgcolor);
      })
      $("#form1 input").on("change",function(){//单选值变化
        var vName=$(this).attr("name");//变量名

        //遍历取消背景显示
        $("#form1 input[name=" + vName + "]").each(function(){
          $(this).parent("label").css("background-color","")
        })
        $(this).parent("label").css("background-color",chkBgcolor);
        //显示序列化值
        var obj1 = $('#form1').serialize();
        $("#p1").text(obj1);
      })
    })
  </script>
</head>
<body>
  <form action="" method="get" id="form1">
    <div>
      时代 :
      <label><input type="radio" name="age" value="1990" checked>1990</label>
    </div>
  </form>
</body>
</html>

```

```

        <label><input type="radio" name="age" value="1991">1991</label>
        <label><input type="radio" name="age" value="1992">1992</label>
    </div>
    <div>
        性别：
        <label><input type="radio" name="sex" value="man" checked>男</label>
        <label><input type="radio" name="sex" value="woman">女</label>
    </div>
</form>
<p id="p1"></p>
</body>
</html>

```

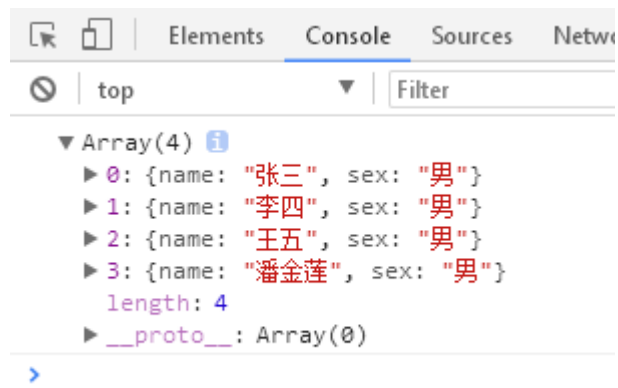
## JS数组筛选法进行json数据对象的查询实现

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <script src="https://cdn.bootcss.com/jquery/3.4.1/jquery.min.js"></script>
    <script>
        $(function(){
            //对象数组
            var datas = [
                {name:"张三",sex:"男"},
                {name:"李四",sex:"男"},
                {name:"王五",sex:"男"},
                {name:"潘金莲",sex:"女"}
            ];
            //筛选
            var nDt = datas.filter(function(peop){
                return peop.sex=="男";
            })
            console.log(nDt);
        })
    </script>
</head>
<body>
    <p id="p1"></p>
</body>
</html>

```

上面的代码筛选出所有男性数据返回一个新数组；（这与传统SQL查询几乎没有区别）



充分利用上面的几点，几乎就可以做到基于json存储、查询数据的基本应用。