[第二章 基础选择器 1](#_Toc410502162)

[2-1#id 选择器(使用身份证号来找人) 1](#_Toc410502163)

[2-2element 选择器(寻找铅笔) 2](#_Toc410502164)

[2-3.class 选择器(寻找红色铅笔) 2](#_Toc410502165)

[2-4\* 选择器(取走全部铅笔) 3](#_Toc410502166)

[2-5sele1,sele2,seleN选择器 3](#_Toc410502167)

[2-6ance desc选择器 4](#_Toc410502168)

[2-7parent > child选择器 4](#_Toc410502169)

[2-8prev + next选择器 4](#_Toc410502170)

[第3章 jQuery过滤性选择器 5](#_Toc410502171)

[3-1:first过滤选择器 5](#_Toc410502172)

[3-2:eq(index)过滤选择器 6](#_Toc410502173)

[3-3:contains(text)过滤选择器 6](#_Toc410502174)

[3-4:has(selector)过滤选择器 7](#_Toc410502175)

[3-5:hidden过滤选择器 8](#_Toc410502176)

[3-6:visible过滤选择器 8](#_Toc410502177)

[3-7[attribute=value]属性选择器 8](#_Toc410502178)

[3-8[attribute!=value]属性选择器 9](#_Toc410502179)

[3-9[attribute\*=value]属性选择器 10](#_Toc410502180)

[3-10 :first-child子元素过滤选择器 10](#_Toc410502181)

[3-11 :last-child子元素过滤选择器 11](#_Toc410502182)

[第4章 jQuery表单选择器 11](#_Toc410502183)

[4-1:input表单选择器 11](#_Toc410502184)

[4-2:text表单文本选择器（空格） 12](#_Toc410502185)

[4-3:password表单密码选择器 12](#_Toc410502186)

[4-4:radio单选按钮选择器 13](#_Toc410502187)

[4-5:checkbox复选框选择器 13](#_Toc410502188)

$(" div ").html("Hello iMooc!");

$(" p").html("Hello iMooc!");

# 第二章 基础选择器

## 2-1#id 选择器(使用身份证号来找人)

过一个id号去查找一个元素，就可以使用如下格式的选择器：

$("#my\_id")

在id号为“default”的元素中显示id号为“divtest”元素的内容。 $("#default").html( $("#devtest").html());

## 2-2element 选择器(寻找铅笔)

，jQuery中可以根据元素名查找元素，格式如下：

$(“element”)

其中element就是元素的名称，

如下图所示：在页面中，根据元素名找到了一个元素，并将它显示的字体加粗。

[](http://img.mukewang.com/5286ddff0001239802840097.jpg)

通过元素名称查找按钮，并通过使用attr()方法将它设置为不可用，其中attr()方法的功能是设置或获取元素的某项属性值。

## 2-3.class 选择器(寻找红色铅笔)

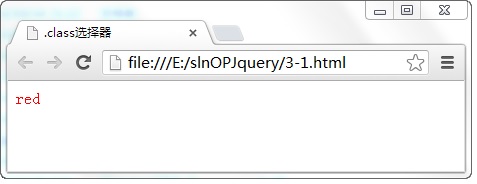
$(“.class”)

其中.class参数表示元素的CSS类别(类选择器)名称。

例如，在页面中，通过class选择器获取某个元素，并显示该元素的class类别名称，如下图所示：

[](http://img.mukewang.com/5286e9f00001a50d03680173.jpg)

在浏览器中显示的效果：

[](http://img.mukewang.com/5286ea780001d74604770184.jpg)

从图中可以看出，通过class选择器的方式获取元素，并调用元素的attr()方法获取元素的类别名称，并将名称显示在页面中。

## 2-4\* 选择器(取走全部铅笔)

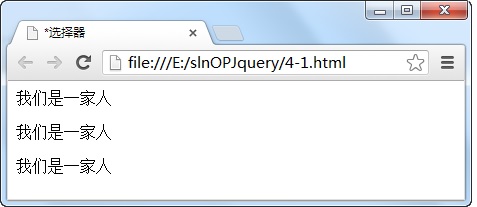
选择器中的参数就一个“\*”，既没有“#”号，也没有“.”号。 由于该选择器的特殊性，它常与其他元素组合使用，表示获取其他元素中的全部子元素。

如下图所示： 使用\*选择器，获取div中的所有子元素并设置三个子元素显示相同的内容。

[](http://img.mukewang.com/5286f46a0001f6dc02890170.jpg)

在浏览器中显示的效果：

由于三个元素都包含在<div>元素中，因此，它们都是<div>元素的子元素，那么，就可以使用$(“div \*”)的方式获取<div>元素中的这三个子元素，并使用html()方法来设置它们显示的内容。

[](http://img.mukewang.com/5286f4a00001331904770207.jpg)

## 2-5sele1,sele2,seleN选择器

有时需要精确的选择任意**多个指定的元素**，类似于从文具盒中挑选出多根自已喜欢的笔，就需要调用sele1,sele2,seleN选择器，它的调用格式如下：

$(“sele1,sele2,seleN”)

其中参数sele1、sele2到seleN为有效选择器，每个选择器之间用“，”号隔开，它们可以是之前提及的各种类型选择器，如$(“#id”)、$(“.class”)、$(“selector”)选择器等。

例如，通过选择器获取其中的任意两个元素，并将它们显示的内容设为相同，如图所示：

[](http://img.mukewang.com/528705b000012a1102890134.jpg)

<script type="text/javascript">

$(".red,.green，.blue").html("hi,我们的样子很美哦!");

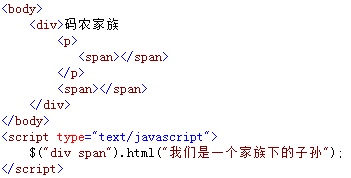
</script>

## 2-6ance desc选择器

ance参数（ancestor祖先的简写）表示父元素；desc参数（descendant后代的简写）表示后代元素，。

比如家族姓氏“div”，家族几代人里，都有名字里带“span”的，就可以用这个ance desc选择器把这几个人给定位出来。

例如，使用层次选择器，获取<div>元素中的全部<span>元素，并设置它们显示的内容，在如下图所示：

[](http://img.mukewang.com/528b4ecd00013c1503510178.jpg)

## 2-7parent > child选择器

与上一节介绍的ance desc选择器相比，parent > child选择器的范围要小些，它所选择的目标是子集元素，相当于一个家庭中的子辈们，但不包括孙辈，它的调用格式如下：

$(“parent > child”)

child参数获取的元素都是parent选择器的子元素，它们之间通过“>”符号来表示一种层次关系。

$("div>label").css("border", "solid 5px red");

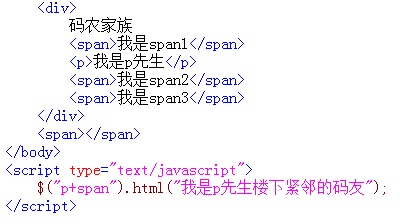
## 2-8prev + next选择器

通过prev + next选择器就可以查找与“prev”元素紧邻的下一个“next”元素，格式如下：

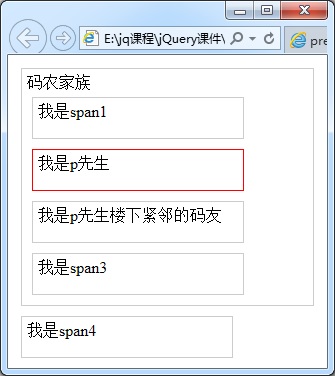
$(“prev + next”)

其中参数prev为任何有效的选择器，参数“next”为另外一个有效选择器，它们之间的“+”表示一种上下的层次关系，也就是说，“prev”元素最紧邻的下一个元素由“next”选择器返回的并且只返回唯的一个元素。

例如，使用prev + next选择器，获取<p>元素最近邻的下一个元素，如下图所示：

[](http://img.mukewang.com/5292e68c000191c204000224.jpg)

在浏览器中显示的效果：

[](http://img.mukewang.com/5292ebe60001dc2b03350376.jpg)

# ****第3章 jQuery过滤性选择器****

## 3-1:first过滤选择器

<script type="text/javascript">

$("li:first").css("background-color", "red");

</script>

## 3-2:eq(index)过滤选择器

如果想从一组标签元素数组中，灵活选择任意的一个标签元素，我们可以使用

:eq(index)

其中参数index表示索引号（即：一个整数），它从0开始，如果index的值为3，表示选择的是第4个元素。例如：

[](http://img.mukewang.com/529c00d40001669503040210.jpg)

在浏览器中显示的效果：

[](http://img.mukewang.com/529c00f90001098804890294.jpg)

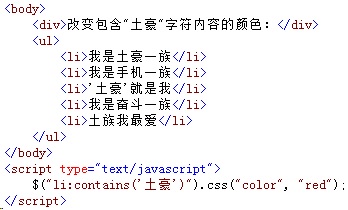
从图中可以看出，通过调用$("li:eq(3)")过滤选择器代码，获取了第4个<li>元素，并使用css()方法设置了该元素在页面中显示的文字样式。

$("li:eq(2)").css("background-color", "#60F");

## 3-3:contains(text)过滤选择器

:contains(text)选择器会更加方便， 它的功能是选择**包含**指定字符串的全部元素，它通常与其他元素结合使用，获取包含“text”字符串内容的全部元素对象。其中参数text表示页面中的文字。

例如:

[](http://img.mukewang.com/529c067b00014cad03480210.jpg)

在浏览器中显示的效果：

[](http://img.mukewang.com/529c069b0001879204890294.jpg)

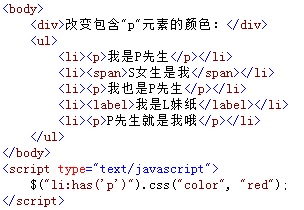
**注意：li:contains('土豪') 土豪为什么必须加单引号呢？因为它是一个字符串，而不是一个变量，所以不加单或双引号的话是会报错的。**

**$("li:contains('jQuery')").css("background", "green");**

## 3-4:has(selector)过滤选择器

除了在上一小节介绍的使用包含的字符串内容过滤元素之外，还可以使用包含的元素名称来过滤，:has(selector)过滤选择器的功能是获取选择器中包含指定元素名称的全部元素，其中selector参数就是包含的元素名称，是被包含元素。

例如：获取指定包含某个元素名的全部<li>元素，并改变它们显示文字的颜色，如下图所示：

[](http://img.mukewang.com/529c1a8300017dd003040210.jpg)

**$("li:has('label')").css("background-color", "blue");**

## 3-5:hidden过滤选择器

:hidden过滤选择器的功能是获取全部不可见的元素，这些不可见的元素中包括type属性值为hidden的元素。

例如，调用:hidden选择器获取不可见的<p>元素，并将该元素的内容显示在<div>元素中

**var $strHTML = $("#hidstr:hidden").val();**

**$("div").html($strHTML);**

**var $strHTML = $("input:hidden").val();**

## 3-6:visible过滤选择器

与上一节的:hidden过滤选择器相反，:visible过滤选择器获取的是全部可见的元素，也就是说，只要不将元素的display属性值设置为“none”，那么，都可以通过该选择器获取。

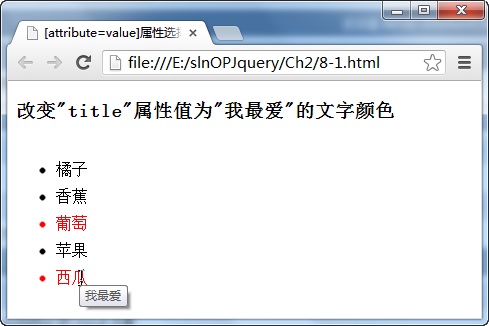
**$("li:visible").css("background-color","blue");**

## 3-7[attribute=value]属性选择器

例如，使用[attribute=value]属性选择器，获取指定属性名和对应值的全部<li>元素，并设置它们显示的文字颜色，如图所示：

[](http://img.mukewang.com/52bcf22e0001409b03620209.jpg)

在浏览器中显示的效果：

[](http://img.mukewang.com/52bcf2560001ad0d04890326.jpg)

从图中可以看出，使用$("li[title='我最爱']")属性选择器代码，获取了2个<li>元素，并调用css()方法设置它们在页面中显示的文字颜色，另外，属性值中的‘’单引号可以不写，由于属性名与属性值是等号，因此，它们之间不是包含关系，而是完全相同。

**$("li[title='蔬菜']").css("background-color", "green");**

## 3-8[attribute!=value]属性选择器

与上一节介绍的[attribute=value]属性选择器正好相反，[attribute!=value]属性选择器的功能是获取不包含属性名，或者与属性名和属性值不相同的全部元素，其中[]是专用于属性选择器的括号符，参数attribute表示属性名称，value参数表示属性值。

**$("li[title!='蔬菜']").css("background-color", "green");**

## 3-9[attribute\*=value]属性选择器

介绍一个功能更为强大的属性选择器[attribute\*=value]，它可以获取属性值中包含指定内容的全部元素，其中[]是专用于属性选择器的括号符，参数attribute表示属性名称，value参数表示对应的属性值。

例如，使用[attribute\*=value]属性选择器，获取属性值中包含某一指定内容的全部<li>元素，并设置它们显示的文字颜色，如下图所示：

$("li[title\*='果']").css("background-color", "green");

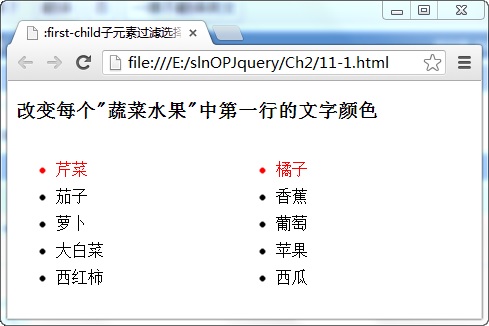
## 3-10 :first-child子元素过滤选择器

通过上面章节的学习，我们知道使用:first过滤选择器可以获取指定父元素中的首个子元素，但该选择器返回的只有一个元素，并不是一个集合，而使用:first-child子元素过滤选择器则可以获取每个父元素中返回的首个子元素，它是一个集合，常用多个集合数据的选择处理。

如下图，如果想把页面中每个ul中的第一个li获取到，并改变其颜色。则可以使用

[](http://img.mukewang.com/529c2a9000014a0e03500322.jpg)

在浏览器中显示的效果：

[](http://img.mukewang.com/529c2ab600011b5804890326.jpg)

通过$("li:first-child")选择器代码，获取了两个<ul>父元素中的第一个<li>元素，并使用css()方法修改了它们在页面中显示的文字颜色。

**$("li:first-child").css("background-color", "green");**

## 3-11 :last-child子元素过滤选择器

与:first-child子元素过滤选择器功能相反，:last-child子元素过滤选择器的功能是获取每个父元素中返回的最后一个子元素，它也是一个集合，常用多个集合数据的选择处理。

**$("li:last-child").css("background-color", "blue");**

# ****第4章 jQuery表单选择器****

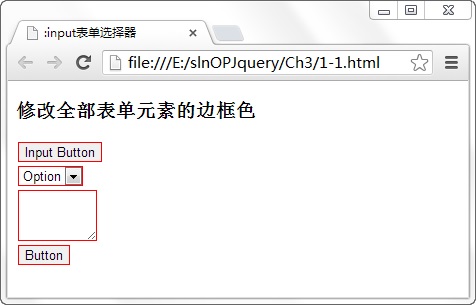
## 4-1:input表单选择器

如何获取表单全部元素？:input表单选择器可以实现，它的功能是返回**全部的表单元素**，不仅包括所有<input>标记的表单元素，而且还包括<textarea>、<select> 和 <button>标记的表单元素，因此，它选择的表单元素是最广的。

如下图所示，使用:input表单选择器获取表单元素，并向这些元素增加一个CSS样式类别，修改它们在页面中显示的边框颜色。



在浏览器中显示的效果：



可以看出，通过调用$("#frmTest :input")表单选择器代码获取了表单中的全部元素，并使用addClass()方法修改它们在页面中显示的边框颜色。addClass()方法的功能是为元素添加指定的样式类别名称，它的更多使用将会在后续章节中进行详细介绍。

## 4-2:text表单文本选择器（空格）

:text表单文本选择器可以获取表单中全部单行的文本输入框元素，单行的文本输入框就像一个不换行的字条工具，使用非常广泛。

例如，在表单中添加多个元素，使用:text选择器获取单行的文本输入框元素，并修改字的边框颜色，如下图所示：



$("#frmTest :text").addClass("bg\_blue"); 有空格

## 4-3:password表单密码选择器

如果想要获取密码输入文本框，可以使用:password选择器，它的功能是获取表单中全部的密码输入文本框元素。

例如，在表单中添加多个输入框元素，使用:password获取密码输入文本框元素，并修改它的边框颜色，如下图所示：



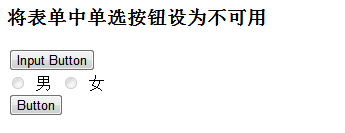
$("form :password").addClass("bg\_red");

$("#frmTest :password").addClass("bg\_red");

## 4-4:radio单选按钮选择器

表单中的单选按钮常用于多项数据中仅选择其一，而使用:radio选择器可轻松获取表单中的**全部单选按钮**元素

$("#frmTest :radio").attr("disabled","true");



## 4-5:checkbox复选框选择器

表单中的复选框常用于多项数据的选择，使用:checkbox选择器可以快速定位并获取表单中的复选框元素。

例如，在表单中增加多个不同类型的元素，使用:checkbox选择器获取其中的全部复选框元素，并将它们全部设为选中状态，如下图所示：

[](http://img.mukewang.com/52971ac700017fbf03830322.jpg)