# 什么是 JQuery

JQuery:Javascript+Query,使用 JavaScript 编写的 js 库,提高开发的效率

# JQuery 特点

1. 轻量级框架:write less,do more 写的少,做的多
2. 应用广泛
3. 免费开源,文档丰富
4. 可以辅助编写众多的第三方库

# 3. 入门程序

步骤:

1. 引入 jquery.js
2. 使用

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<!-- 1.引入 jquery -->

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(document).ready(function(){ alert("Hello World,JQuery!");

});

</script>

</head>

<body>

</body>

</html>

# JQuery 语法

## 语法

$(“选择器”).action()

## 说明

$:核心函数

选择器:JQuery 查找网页元素的表达式,返回的是 jquery 对象

action:jquery 内置函数,实现特定的功能

## 核心函数的功能

1. 获取 dom 对象
2. 创建 dom 对象
3. 克隆 dom 对象

# 什么是 Jquery 对象

使用 jQuery 封装的 Dom 对象叫做 jQuery 对象

# 使用 jQuery 对象的优缺点：

* 优点：jQuery 的扩展属性和函数在底层判断了浏览器以及版本实现了浏览器的兼容。
* 缺点：jQuery 扩展的函数只能实现基本的 dom 操作，但是对于元素独有的 htmldom 接口封装很少(例如 input、select())，对于 html5 新增的 DOM 方法也不支持，如果想实现类似功能，只能使用 dom 接口。

结论：为保证兼容性尽量使用 jQuery 对象的 API，但特殊功能也需要使用原生 dom 接口。

# 编码习惯

如果获取的是 jQuery 对象,那么要在变量前面加上$。

* var $variable=jQuery 对象
* var variable=DOM 对象

# DOM 转 Jquery 对象

## 语法

var $jquery 变量=$(dom 对象);

## 案例

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

// 使用 js 给 div 赋值

/\* function dian(){

//1.得到 div 的 dom 对象

var div1=document.getElementById("div1");

//2.赋值

div1.innerHTML="我是 div 的内容";

} \*/

//使用 jquery 给 div 赋值

/\* function dian(){

//1.得到 div 的 jquery 对象

var $div=$("#div1");

//2.赋值

$div.html("我是 Jquery 的 div 的内容");

} \*/

//Dom 对象转换为 jquery 对象

function dian(){

//1.得到 div 的 dom 对象

var div1=document.getElementById("div1");

//1.使用$(dom 对象)得到 div 的 jquery 对象

var $div=$(div1);

//2.赋值

$div.html("我是 Jquery 的 div 的内容");

}

</script>

</head>

<body>

<div id="div1">

</div>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

1. **JQuery 转 Dom 对象**
   1. **语法**
2. jquery 对象[index]
3. jquery 对象.get(index)

## 案例

//Jquery 对象转为 dom 对象

function dian(){

//1.得到 div 的 jquery 对象

var $div=$("#div1");

//2.1.使用 jquery 对象[index]

// var div=$div[0];

//2.2.使用 jquery 对象.get(index) var div=$div.get(0);

//3.赋值

div.innerHTML="我是 div 的内容";

}

<body>

<div id="div1">

</div>

<button type="button" onclick="dian()">点我</button>

</body>

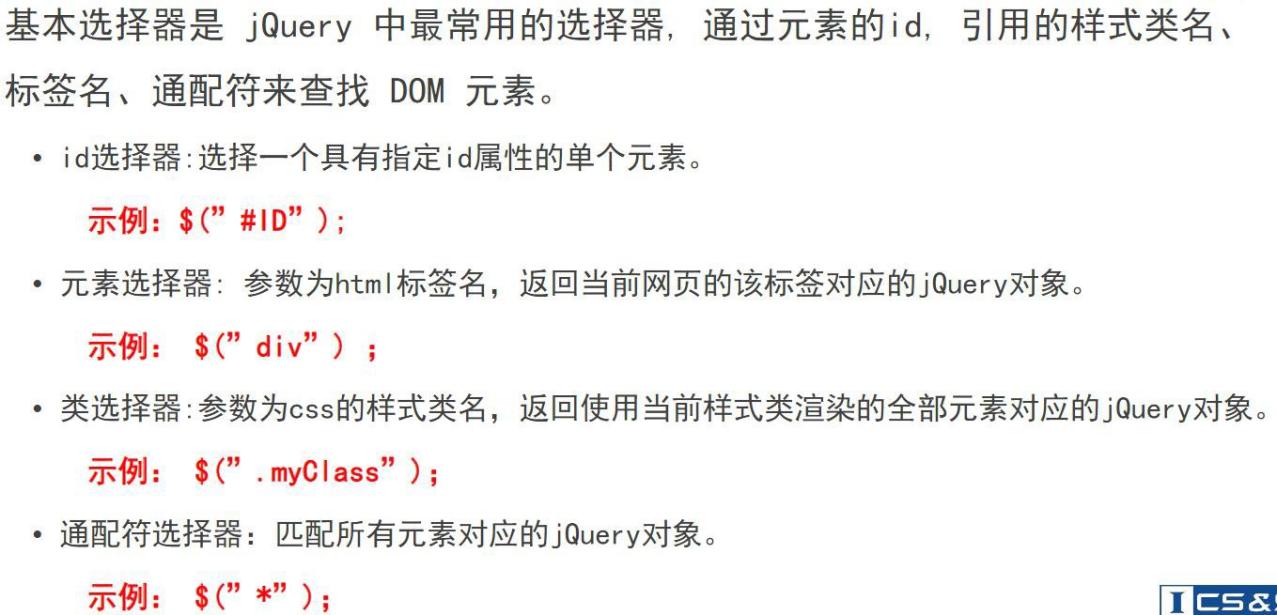
# 什么是 jQuery 选择器

用于获取 jQuery 对象的表达式

# jQuery 选择器的基本语法

$(“选择器表达式”)

# 基本选择器



组合(合并)选择器语法:

$(选择器 1,选择器 2,选择器 3,...) 或的关系(合并)

## id 选择器

$(“#id”)

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("#myDiv").css("background","red");

}

</script>

</head>

<body>

<div id="notMe"><p>id="notMe"</p></div>

<div id="myDiv">id="myDiv"</div>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 元素选择器

$(“元素名”)

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("div").css("background","red");

}

</script>

</head>

<body>

<div>

div1

</div>

<div>

div2

</div>

<p>p</p>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 类选择器

$(“.类名”)

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$(".myClass").css("background","red");

}

</script>

</head>

<body>

<div class="notMe">div class="notMe"</div>

<div class="myClass">div class="myClass"</div>

<span class="myClass">span class="myClass"</span>

<br/>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 通配符选择器

$(“\*”) 匹配所有的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("\*").css("background","red");

}

</script>

</head>

<body>

<div>DIV</div>

<span>SPAN</span>

<p>P</p>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 组合(合并)选择器

语法:

$(“选择器 1,选择器 2,选择器 3,...”) 或的关系(合并) 将每一个选择器匹配到的元素合并后一起返回。你可以指定任意多个选择器，并将匹配到的元素合并到一个结果内

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("div,span,p.myClass").css("background","red");

}

</script>

</head>

<body>

<div>div</div>

<p class="myClass">p class="myClass"</p>

<span>span</span>

<p class="notMyClass">p class="notMyClass"</p>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

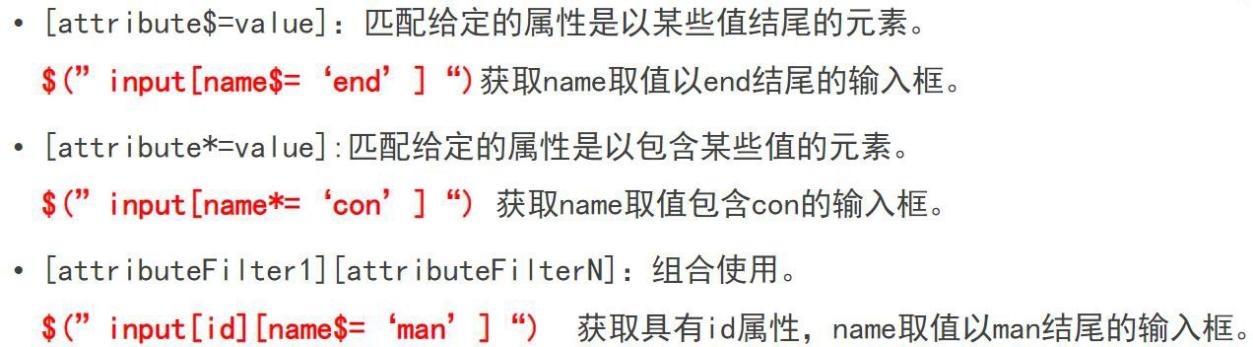
# 属性选择器

## 定义

通过元素的属性获取相应的元素

## 分类





* 1. **[attribute]**

匹配包含给定属性的元素,属性有没有值无所谓.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("div[id]").css("background","red");

}

</script>

</head>

<body>

<div>

<p>Hello!</p>

</div>

<div id="test2">div</div>

<div id>div2</div>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## [attribute=value]

匹配给定的属性是某个特定值的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("input[name='newsletter']").attr("checked",true);

}

</script>

</head>

<body>

<input type="checkbox" name="newsletter" value="Hot Fuzz" />Hot Fuzz

<input type="checkbox" name="newsletter" value="Cold Fusion" />Cold Fusion

<input type="checkbox" name="accept" value="Evil Plans" />Evil Plans<br/>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## [attribute!=value]

匹配所有不含有指定的属性，或者属性不等于特定值的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("input[name!='newsletter']").prop("checked",true);

}

</script>

</head>

<body>

<input type="checkbox" name="newsletter" value="Hot Fuzz" />Hot

Fuzz

<input type="checkbox" name="newsletter" value="Cold Fusion"

/>Cold Fusion

<input type="checkbox" name="accept" value="Evil Plans" />Evil

Plans

<input type="checkbox" value="Evil Plans2" />Evil Plans2

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## [attribute^=value]

匹配给定的属性是以某些值开始的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("input[name^='news']").css("background","red");

}

</script>

</head>

<body>

<input name="newsletter" />

<input name="milkman" />

<input name="newsboy" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## [attribute$=value]

匹配给定的属性是以某些值结尾的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("input[name$='letter']").css("background","red");

}

</script>

</head>

<body>

<input name="newsletter" />

<input name="milkletter" />

<input name="newsboy" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## [attribute\*=value]

匹配给定的属性是以包含某些值的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("input[name\*='let']").css("background","red");

}

</script>

</head>

<body>

<input name="newsletter" />

<input name="milkletter" />

<input name="newsboy" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 复合属性选择器

* + 1. **语法**

[selector1][selector2][selector3]...

复合属性选择器，需要同时满足多个条件时使用

## 案例

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("input[id][name$='man']").css("background","red");

}

</script>

</head>

<body>

<input id="man-news" name="man-news" />

<input name="milkman" />

<input id="letterman" name="new-letterman" />

<input name="newmilk" />

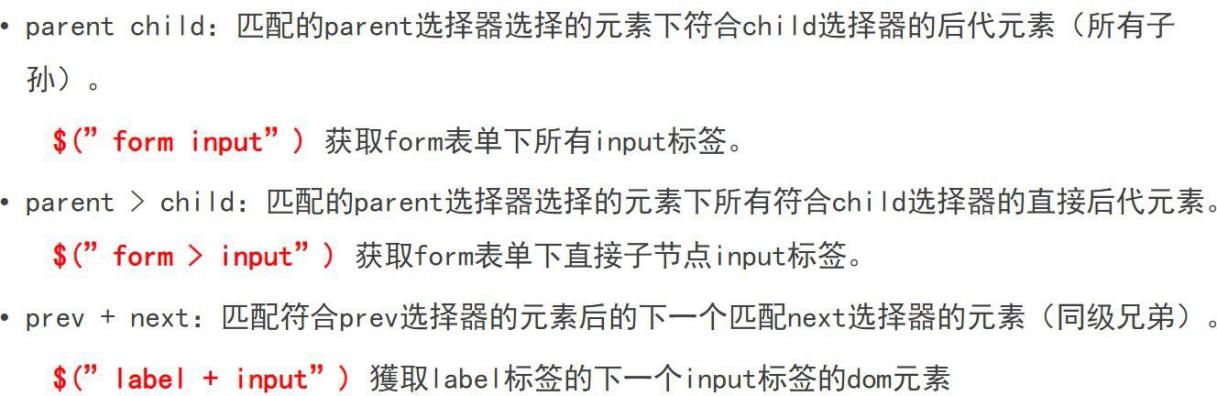
<button type="button" onclick="dian()">点我</button>

</body>

</html>

1. **层次选择器：**

通过 DOM 元素之间的层次关系来获取特定元素,例如后代元素,子元素,相邻元素,兄弟元素等



## 后代选择器

使用空格,在给定的祖先元素下匹配所有的后代元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

// 选择 form 的 input 后代

$("form input").css("background","red");

}

</script>

</head>

<body>

<form>

<label>Name:</label>

<input name="name" />

<fieldset>

<label>Newsletter:</label>

<input name="newsletter" />

</fieldset>

</form>

<input name="none" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 子元素选择器

使用>,在给定的父元素下匹配所有的子元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

// 选择 form 的 input 后代

$("form > input").css("background","red");

}

</script>

</head>

<body>

<form>

<label>Name:</label>

<input name="name" />

<fieldset>

<label>Newsletter:</label>

<input name="newsletter" />

</fieldset>

</form>

<input name="none" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 相邻元素选择器

使用+隔开, 匹配所有紧接在 prev 元素后的 next 元素(如果有,是一个元素)

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

// 选择紧邻 label 的 input

$("label + input").css("background","red");

}

</script>

</head>

<body>

<form>

<label>Name:</label>

<input name="name" />

<fieldset>

<label>Newsletter:</label>

<input name="newsletter" />

</fieldset>

</form>

<input name="none" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## 匹配 prev 元素之后的所有 siblings 元素

prev ~ siblings:匹配 prev 元素之后的所有 siblings 元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

// 选择 label 后的所有 input

$("label ~ input").css("background","red");

}

</script>

</head>

<body>

<form>

<input name="name5" />

<input name="name6" />

<label>Name:</label>

<input name="name" />

<input name="name2" />

<input name="name3" />

<fieldset>

<label>Newsletter:</label>

<input name="newsletter" />

<input name="newsletter2" />

</fieldset>

</form>

<input name="none" />

<button type="button" onclick="dian()">点我</button>

</body>

</html>

# 基础过滤选择器



## :first

匹配找到的第一个元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

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

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## :last

匹配找到的最后一个元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("li:last").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## :not

返回不匹配指定选择器的元素.

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("li:not(:last)").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## :even

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("li:even").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

* 1. **:eq(index)**

选择索引=index 的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("li:eq(2)").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

function dian(){

$("li:gt(2)").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$("li:lt(2)").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>list item 1</li>

<li>list item 2</li>

<li>list item 3</li>

<li>list item 4</li>

<li>list item 5</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

## :header

匹配 h1~h6 的标题元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

$(":header").css("background","red");

}

</script>

</head>

<body>

<h1>Header 1</h1>

<p>Contents 1</p>

<h2>Header 2</h2>

<p>Contents 2</p>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

# 子元素过滤选择器



## :nth-child

匹配其父元素下的第 N 个子或奇偶元素

':eq(index)' 只匹配一个元素，而这个将为每一个父元素匹配子元素。:nth-child 从 1 开始的，而:eq() 是从 0 算起的！可以使

用:<br>nth-child(even)<br>:nth-child(odd)<br>:nth-child(3n)<br>:nth-child(2)<br>:nth

-child(3n+1)<br>:nth-child(3n+2)

<!DOCTYPE html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

// $("ul li:nth-child(2)").css("background","red");

// $("ul li:nth-child(even)").css("background","red");

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

$("ul li:nth-child(3n+2)").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>John</li>

<li>Karl</li>

<li>Brandon</li>

<li>Brandon</li>

<li>Brandon</li>

</ul>

<ul>

<li>Glen</li>

<li>Tane</li>

<li>Ralph</li>

<li>Ralph</li>

</ul>

<button type="button" onclick="dian()">点我</button>

</body>

</html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function dian(){

// $("ul li:nth-child(2)").css("background","red");

// $("ul li:nth-child(even)").css("background","red");

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

// $("ul li:nth-child(3n+2)").css("background","red");

// $("ul li:first-child").css("background","red");

// $("ul li:last-child").css("background","red");

$("ul li:only-child").css("background","red");

}

</script>

</head>

<body>

<ul>

<li>John</li>

<li>Karl</li>

<li>Brandon</li>

<li>Brandon</li>

<li>Brandon</li>

</ul>

<ul>

<li>Glen</li>

<li>Tane</li>

<li>Ralph</li>

<li>Ralph</li>

</ul>

<ul>

<li>Glen</li>

</ul>

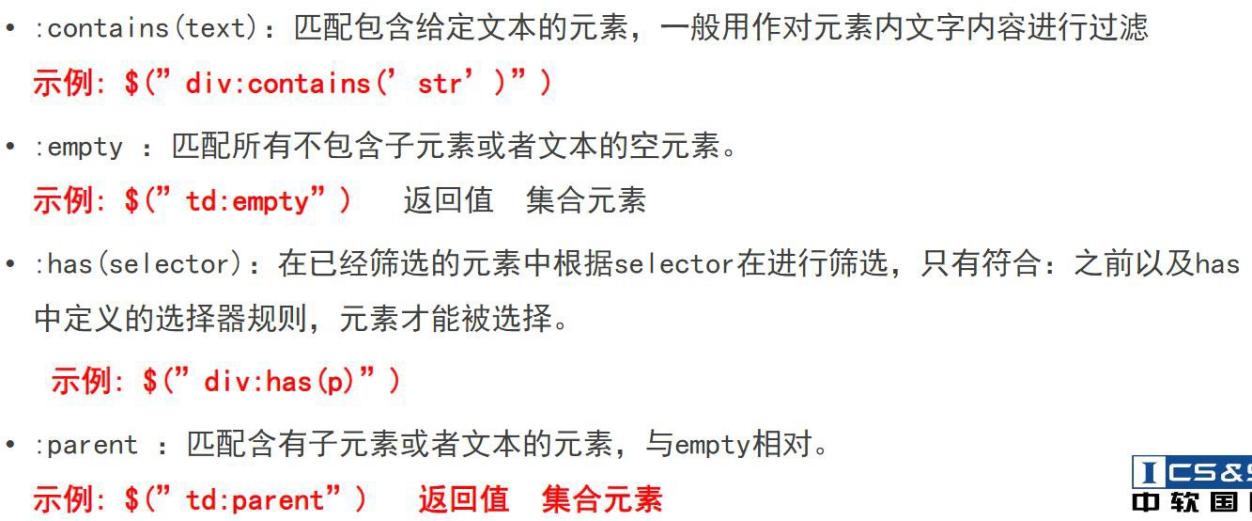
<button type="button" onclick="dian()">点我</button>

</body>

</html>

# 内容过滤选择器

## 定义

在已经选择的对象中根据它所包含的子元素和文本内容进行过滤。

## :contains(text)

匹配包含给定文本的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//查找内容包含 Jhon 的 div 元素

$("div:contains('John')").css("background","red");

}

</script>

</head>

<body>

<div>John Resig</div>

<div>George Martin</div>

<div>Malcom John Sinclair</div>

<div>J. Ohn </div>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :empty

匹配所有不包含子元素或者文本的空元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//查找 td 为空的元素

$("td:empty").css("background","red");

}

</script>

</head>

<body>

<table>

<tr><td>Value 1</td><td></td></tr>

<tr><td>Value 2</td><td></td></tr>

</table>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :has(selector)

匹配含有选择器所匹配的元素的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//has(selector):查找含有 p 标签元素的 div 元素

$("div:has(p)").css("background","red");

}

</script>

</head>

<body>

<div><p>Hello</p></div>

<div>Hello again!</div>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :parent()

匹配含有子元素或者文本的元素，与 empty 相对

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//查找 td 为空的元素

// $("td:empty").css("background","red");

//查找匹配含有子元素或者文本的元素，与 empty 相对

$("td:parent").css("background","red");

}

</script>

</head>

<body>

<table>

<tr><td>Value 1</td><td></td></tr>

<tr><td>Value 2</td><td></td></tr>

</table>

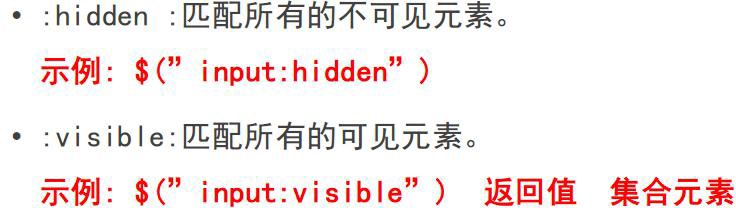
<button type="button" onclick="test()">点我</button>

</body>

</html>

# 可见性过滤选择器

是根据元素的可见和不可见状态来选择相应的元素



## :hidden

匹配所有不可见元素，或者 type 为 hidden 的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//查找隐藏的<tr></tr>

$("tr:hidden").css("background","red").show();//.show()显示 .hide()不显示var val=$("input:hidden").val();

console.log(val);

}

</script>

</head>

<body>

<table>

<tr style="display:none"><td>Value 1</td></tr>

<tr><td>Value 2</td></tr>

</table>

<form>

<input type="text" name="email" />

<input type="hidden" name="id" value="123456"/>

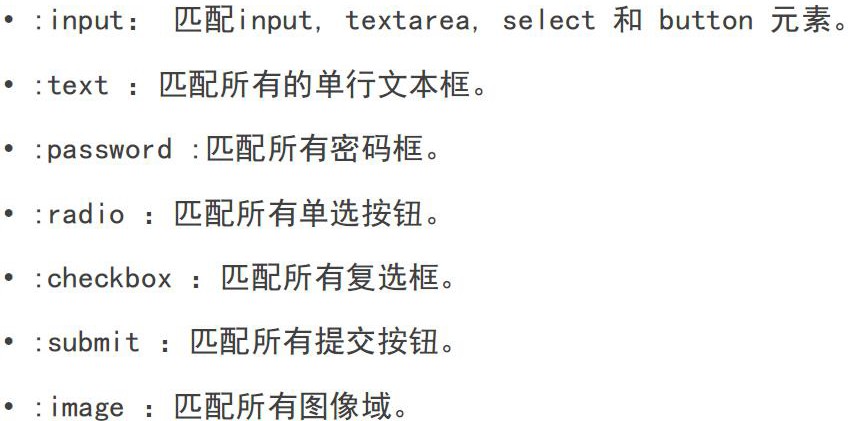
</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# 表单类型过滤器



## :input

匹配所有 input, textarea, select 和 button 元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

$(":input").css("background","red");

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :text

匹配所有的单行文本框

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

// $(":input").css("background","red");

//匹配所有的单行文本框

$(":text").css("background","red");

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :password

匹配所有密码框

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

// $(":input").css("background","red");

//匹配所有的单行文本框

// $(":text").css("background","red");

//:password :匹配所有密码框

$(":password").css("background","red");

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :radio

匹配所有单选按钮

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

// $(":input").css("background","red");

//匹配所有的单行文本框

// $(":text").css("background","red");

//:password :匹配所有密码框

// $(":password").css("background","red");

//:radio ：匹配所有单选按钮

$(":radio").prop("checked",true);

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :checkbox

匹配所有复选框

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

// $(":input").css("background","red");

//匹配所有的单行文本框

// $(":text").css("background","red");

//:password :匹配所有密码框

// $(":password").css("background","red");

//:radio ：匹配所有单选按钮

// $(":radio").prop("checked",true);

//:checkbox ：匹配所有复选框

$(":checkbox").prop("checked",true);

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :submit

匹配所有提交按钮

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

// $(":input").css("background","red");

//匹配所有的单行文本框

// $(":text").css("background","red");

//:password :匹配所有密码框

// $(":password").css("background","red");

//:radio ：匹配所有单选按钮

// $(":radio").prop("checked",true);

//:checkbox ：匹配所有复选框

// $(":checkbox").prop("checked",true);

// :submit ：匹配所有提交按钮

$(":submit").css("background","red");

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :image

匹配所有图像域

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript"

charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有 input, textarea, select 和 button 元素

// $(":input").css("background","red");

//匹配所有的单行文本框

// $(":text").css("background","red");

//:password :匹配所有密码框

// $(":password").css("background","red");

//:radio ：匹配所有单选按钮

// $(":radio").prop("checked",true);

//:checkbox ：匹配所有复选框

// $(":checkbox").prop("checked",true);

// :submit ：匹配所有提交按钮

// $(":submit").css("background","red");

// :image ：匹配所有图像域

$(":image").css("background","red");

}

</script>

</head>

<body>

<form>

<input type="button" value="Input Button"/>

<input type="checkbox" />

<input type="file" />

<input type="hidden" />

<input type="image" />

<input type="password" />

<input type="radio" />

<input type="reset" />

<input type="submit" />

<input type="text" />

<select><option>Option</option></select>

<textarea></textarea>

<button>Button</button>

</form>

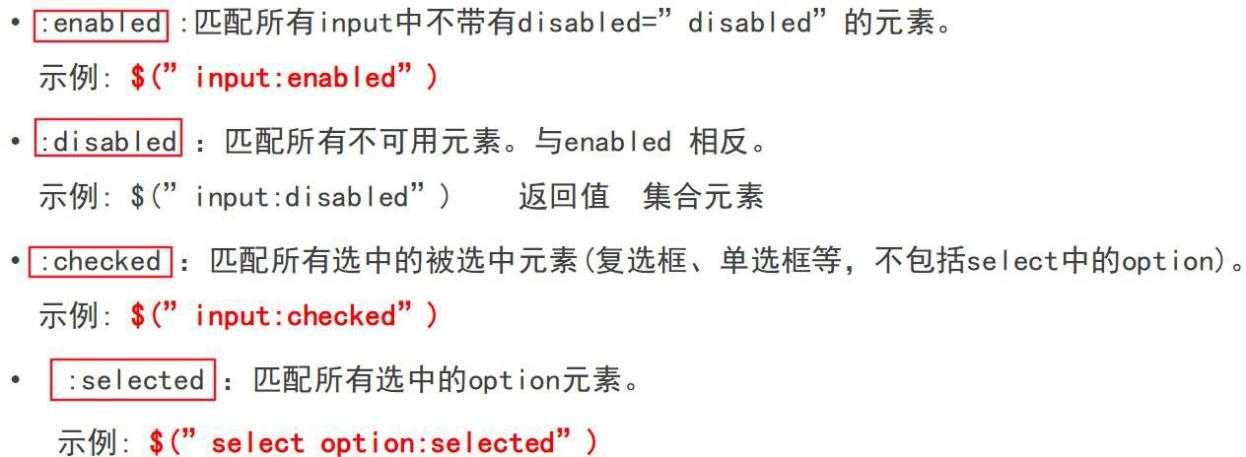
<button type="button" onclick="test()">点我</button>

</body>

</html>

# 表单状态过滤器

状态过滤器：根据表单元素当前状态进行过滤的选择器



## :enabled

匹配所有可用元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//查找所有可用的 input 元素

$("input:enabled").css("background","red");

}

</script>

</head>

<body>

<form>

<input name="email" disabled="disabled" />

<input name="id" />

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :disabled

匹配所有不可用元素。与 enabled 相反

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

function test(){

//查找所有可用的 input 元素

// $("input:enabled").css("background","red");

// 查找所有不可用的 input 元素

$("input:disabled").css("background","red");

}

</script>

</head>

<body>

<form>

<input name="email" disabled="disabled" />

<input name="id" />

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :checked

匹配所有选中的被选中元素(复选框、单选框等，不包括 select 中的 option)。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

// 匹配所有选中的被选中元素(复选框、单选框等，不包括 select 中的

option)

$("input:checked").each(function(){

//this 代表当前被访问的 dom 对象

console.log(this.value);

});

}

</script>

</head>

<body>

<form>

<input type="checkbox" name="newsletter" checked="checked" value="Daily" />

<input type="checkbox" name="newsletter" value="Weekly" />

<input type="checkbox" name="newsletter" checked="checked" value="Monthly" />

</form>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## :selected

匹配所有选中的 option 元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//匹配所有选中的 option 元素

var val=$("select option:selected").val(); console.log(val);

}

</script>

</head>

<body>

<select>

<option value="1">Flowers</option>

<option value="2" selected="selected">Gardens</option>

<option value="3">Trees</option>

</select>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# 创建 DOM 元素

使用 jQuery 的工厂函数 $(html); 会根据传入的 html 标记字符串创建一个 jQuery 的DOM

对象并返回注意:

1. 动态创建的新元素节点不会被自动添加到文档中, 而是需要使用其他方法将其插入到文档中 例如 append（）;
2. 当创建单个元素时, 需注意闭合标签和使用标准的 XHTML 格式. 例如创建一个<p>元素, 可以使用 $(“<p/>”) 或 $(“<p></p>”)。

案例:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//使用 jquery 创建 dom 元素$(html)

var img=$('<img src="img/1.png" alt="图片" style="width: 100px;"/>')

//插入到文档中

$(document.body).append(img);

}

</script>

</head>

<body>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# 23. 插入 dom 元素

将指定的或新创建的 jQuery 的 dom 对象插入到指定位置上。按照插入方式的不同，可分为插入到某元素内作为该元素的子节点方式；

1. parent.append(child) :向父元素（parent）加入子元素（child），插入后子元素为父元素的最后一个孩子。
2. child.appendTo(parent) :将子元素（child）插入到父元素（parent）下，插入后子元素为父

元素的最后一个孩子。与 append 调用方式相反，功能相同。

1. parent.prepend(child):向父元素（parent）加入子元素（child），插入后子元素为父元素的 第一个孩子。
2. child.prependTo(parent) :将子元素（child）插入到父元素（parent）下，插入后子元素为父元素的第一个孩子。与 prepend 调用方式相反，功能相同。

注意:

parent、child 一般的使用场景都是 jQuery 的 dom 对象。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

//使用 jquery 创建 dom 元素$(html)

var img=$('<img src="img/1.png" alt="图片" style="width: 100px;"/>')

//1.插入到文档中:append 插入到父元素内部的结尾处

// $(document.body).append(img);//append 插入到父元素内部的结尾处

//2.插入到文档中:append 插入到父元素内部的结尾处

//img.appendTo($(document.body));

//3.往父元素的第一个位置插入

//$(document.body).prepend(img);

//4 往父元素的第一个位置插入

img.prependTo($(document.body));

}

</script>

</head>

<body>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# **24.**插入到某元素外作该元素的兄弟节点方式

brother.after(obj) :在某个兄弟节点（brother）之后插入目标对象（obj）。brother.before(content):在某个兄弟节点（brother）之前插入目标对象（obj）。obj.insertAfter(content):将目标对象（obj）在插入到某个兄弟节点（brother）之后。obj.insertBefore(content) :将目标对象（obj）在插入到某个兄弟节点（brother）之前。brother、obj 一般的使用场景都是 jQuery 的 dom 对象。

注意:

brother、obj 一般的使用场景都是 jQuery 的 dom 对象

## **24.1.**brother.after(obj)

在某个兄弟节点（brother）之后插入目标对象（obj）。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

$("p").after("<b>hello</b>");

}

</script>

</head>

<body>

<p>I would like to say: </p>

<p>I would like to say: </p>

<p>I would like to say: </p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## **24.2.**brother.before(content):

在某个兄弟节点（brother）之前插入目标对象（obj）。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

// 向每个 p 标签的后面插入 hello

// $("p").after("<b>hello</b>");

// 向每个 p 标签的前面插入 hello

$("p").before("<b>hello</b>");

}

</script>

</head>

<body>

<p>I would like to say: </p>

<p>I would like to say: </p>

<p>I would like to say: </p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## **24.3.**obj.insertAfter(content)

将目标对象（obj）在插入到某个兄弟节点（brother）之后。

$("<b>Hello</b>").insertAfter($("p"));

}

$("p").before("<b>Hello</b>");

//

$("p").after("<b>Hello</b>");

//往 p 标签的前面插

//

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test () {

//往 p 标签的后面插

</script>

</head>

<body>

<p>我想说的是:</p>

<p>我想说的是:</p>

<p>我想说的是:</p>

<button onclick="test()">测试</button>

</body>

</html>

## **24.4.**obj.insertBefore(content)

将目标对象（obj）在插入到某个兄弟节点（brother）之前。

brother、obj 一般的使用场景都是 jQuery 的 dom 对象。

</head>

<body>

<p>我想说的是:</p>

<p>我想说的是:</p>

<p>我想说的是:</p>

<button onclick="test()">测试</button>

$("<b>Hello</b>").insertAfter($("p"));

$("<b>Hello</b>").insertBefore($("p"));

}

</script>

//

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test () {

//往 p 标签的后面插

// $("p").after("<b>Hello</b>");

//往 p 标签的前面插

// $("p").before("<b>Hello</b>");

</body>

</html>

# **26.**dom 元素的克隆

clone([Even[,deepEven]])

创建一个指定的元素集合完全相同的副本。完全相同泛指数据相同，副本指内存地址不同。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

$("b").clone().prependTo($("p"));

}

</script>

</head>

<body>

<b>Hello</b><p>, how are you?</p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# **27.**dom 元素的替换

使用一个元素替换另外一个元素。

obj.replaceAll(‘selector’/dom):使用 obj 替换 dom 对象或 selector 筛选的 dom 对象。（jQuery 的参数建议使用 dom 对象传递）。替换后 obj 被 clone，分别替换目标元素。obj.replaceWith(‘selector’/dom):与 replaceAll 调用方式相反。

## **27.1.**replaceAll(selector)

用匹配的元素替换掉所有 selector 匹配到的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

// 把所有的段落标记替换成加粗标记

$("<b>加粗标签</b>").replaceAll($("p"));

}

</script>

</head>

<body>

<p>Hello</p><p>cruel</p><p>World</p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## **27.2.**replaceWith(content|fn)

将所有匹配的元素替换成指定的 HTML 或 DOM 元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

// 把所有的段落标记替换成加粗标记

// $("<b>加粗标签</b>").replaceAll($("p"));

$("p").replaceWith("<b>加粗标签</b>");

}

</script>

</head>

<body>

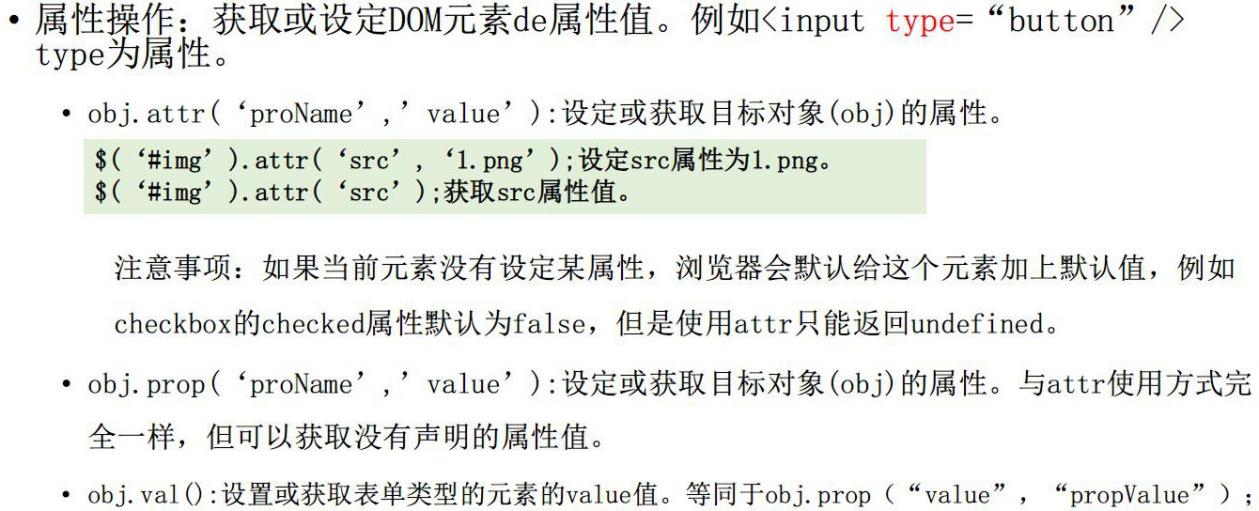
<p>Hello</p><p>cruel</p><p>World</p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# **28.**属性



## **28.1.**课堂案例

使用 prop 设计 table 里的 tr 有 checkbox

|  |  |  |  |
| --- | --- | --- | --- |
| 全选/全不选 复选框 | 序号 | 姓名 | 年龄 |
| 复选框 | 1 | 刘备 | 23 |
| 复选框 | 2 | 关羽 | 22 |
| 复选框 | 3 | 张飞 | 21 |

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function selAll(){

$(".ck").prop("checked",$("#ckAll").prop("checked"));

}

function ckClick(){

$("#ckAll").prop("checked",$(".ck:checked").length==$(".ck").length);

}

</script>

</head>

<body>

<table border="1" cellspacing="0" cellpadding="">

<tr>

/></th>

<th> 全 选 / 全 不 选 <input type="checkbox" id="ckAll" onclick="selAll()"

<th>序号</th>

<th>姓名</th>

<th>年龄</th>

</tr>

<tr>

<td>

<input type="checkbox" class="ck" onclick="ckClick()" />

</td>

<td>

1

</td>

<td>刘备</td>

<td>23</td>

</tr>

<tr>

<td>

<input type="checkbox" class="ck" onclick="ckClick()" />

</td>

<td>

2

</td>

<td>关羽</td>

<td>22</td>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

console.log($("#name").val());

$("#name").val("李四");

}

</script>

</head>

<body>

<input type="text" id="name" value="张三" />

<button type="button" onclick="test()">点我</button>

</body>

</html>

</tr>

<tr>

<td>

<input type="checkbox" class="ck" onclick="ckClick()" />

</td>

<td>

3

</td>

<td>张飞</td>

<td>20</td>

</tr>

</table>

</body>

</html>

# **29.**class 操作：

修改目标元素所使用的样式类。

obj.addClass(‘className’):为目标元素增加样式类（className）。 示例：$(“p”).addClass(“font”);//font 为样式类名

obj.hasClass(‘className’):判断目标元素是否使用指定样式，返回值 boolean。示例：var boolean =$(“p”).hasClass(“font”);

obj.removeClass(‘className’):删除目标元素对样式类（className）的使用。 示例：$(“p”).removeClass(“font”);

obj.toggleClass(‘className’):反转目标元素对样式类（className）的使用。如果已经使用目 标样式则删除，没使用则添加。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

$("p").addClass("p1");

/\*var b=$("p").hasClass("p1"); console.log(b); \*/

// $("p").toggleClass("p1");

$("p").removeClass("p1");

}

</script>

</head>

<body>

<p>刘备</p>

<p>刘备 2</p>

<p>刘备 3</p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

# **30.**样式操作

不通过设定样式类而直接修改样式的方式。

obj.css(‘styleName’):获取目标样式的取值。示例：$('#div').css("font-size")

obj.css(‘styleName’,’styleValue’)：设定目标元素样式的值为指定值： 示例：$('#div').css('color','red');

obj.css({‘styleName’,’styleValue’,..})：批量设定目标元素的样式值。示例：$('#div').css({color:'red','background-color':'yellow'});

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test () {

//使用 css(参数,值)设置 p 标签的字体颜色为红色

$("p").css("color","red");

//使用 css(参数)得到 p 标签的字体颜色的值

console.log($("p").css("color"));

//一次批量设置多个样式属性和值

$("p").css({"font-size":"38px","background-color":"yellow"});

}

</script>

</head>

<body>

<p>p 标签 1</p>

<p>p 标签 2</p>

<p>p 标签 3</p>

<p>p 标签 4</p>

<p>p 标签 5</p>

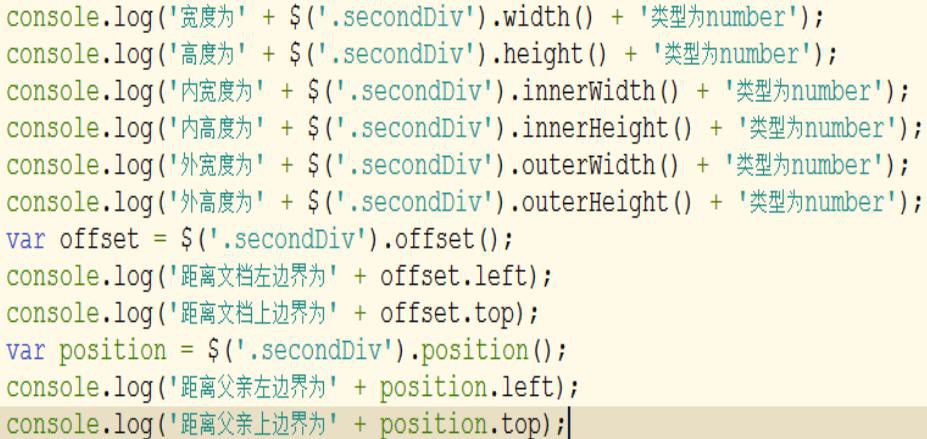
<button onclick="test()">测试</button>

</body>

</html>

# **31.**查询当前元素的大小以及位置

jQuery 封装了一组接口用于查询当前元素的大小以及位置



# **32.**jQuery 对于文本内容的封装。

obj.html():读取和设置某个元素中的 HTML 内容，与 dom 中的 innerHTML 类似。取得第一个匹配元素的 html 内容

var result=$("p").html();

console.log(result);

$("p").html("<b>关羽</b>")

obj.text():读取和设置某个元素中的文本内容，与 dom 中的 innerText 类似。

var result=$("p").text();

console.log(result);

$("p").text("<b>关羽</b>")

# **33.**on和off 方法使用

on 方法绑定事件和处理函数

## **33.1.**语法

$(‘outerSelector’).on(‘eventType’,’childSelector’,data,fun);

## **33.2.**说明

outerSelector:外层区域选择器。eventType：事件类型 如 click，字符类型。

childSelector：内部元素选择器（缺省为外部选择器）。

data：传递给执行函数的参数，封装在 event.data 中。json 类型。

fun：事件处理函数

## **33.3.**案例

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

//on 绑定事件

/\* $("p").on("click",function(){

console.log($(this).text());

}); \*/

//简写

/\* $("p").click(function(){

console.log($(this).text());

}) \*/

//传参

$("p").on("click",{id:1},function(e){ console.log(e.data.id);

});

})

</script>

</head>

<body>

<p>刘备</p>

<p>刘备 2</p>

<p>刘备 3</p>

</body>

</html>

**34.off 方法使用**

off 方法解除事件和处理函数

## **33.3.**语法

$(‘outerSelector’).off(‘eventType’,’childSelector’,fun);

## **33.4.**说明

outerSelector:外层区域选择器。eventType：事件类型 如 click，字符类型

childSelector：内部元素选择器（缺省外部选择器）。fun：需要解除的事件处理函数

## **33.5.**案例

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

//on 绑定事件

function fun(){

console.log($(this).text());

}

$("p").on("click",fun);

//简写

/\* $("p").click(function(){

console.log($(this).text());

}) \*/

//传参

/\* $("p").on("click",{id:1},function(e){ console.log(e.data.id);

}); \*/

//解绑事件

//$("p").off("click",fun);

$("p").off("click");

})

</script>

</head>

<body>

<p>刘备</p>

<p>刘备 2</p>

<p>刘备 3</p>

</body>

</html>

# **35.特殊事件绑定方式(简写)**

为了编写方便，jQuery 定义了特殊事件绑定函数来绑定事件。

$(‘button’).click(function() {});

等价

$(‘button’).on(‘click’,function() {});

特殊事件绑定的方式为：

$(‘selector’).eventType(data,function() {}); selector：选择器。

eventType:函数名，但直接对应事件类型（注意省略 on）。data：传递给执行函数的参数，封装在 event.data 中。json 类型。

$("p").click(function(){

console.log($(this).text());

})

# **36.**窗口事件类型函数

## **36.1.**resize

用于窗口对象（window），当窗口大小发生变化时调用。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$(window).resize(function(){ alert("我被改变大小了!");

});

});

</script>

</head>

<body>

</body>

</html>

## **36.2.**scroll

用于窗口、iframe、div 等可能出现滚动条的对象。当使用鼠标滚轮滚动时调用。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

var count=0;

$(window).scroll(function(){

// alert(count)

$("span").text(count++);

});

});

</script>

</head>

<body>

<div style="height: 2000px;background-color: aquamarine;">

</div>

<span>0</span>

</body>

</html>

window.onload、body.onload、document 的区别：

# **37.**表单事件函数：

## **37.1.**focus

当元素获得焦点时触发目标函数，不支持冒泡；

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

//在页面加载完成后让#username 得到焦点

$("#username").focus();

});

</script>

</head>

<body>

<input type="text" id="username" value="zs" />

</body>

</html>

## **37.2.**focusin

和 focus 一样，只是此事件支持冒泡；focusin 事件跟 focus 事件区别在于，他可以在父元素上检测子元素获取焦点的情况。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("p").focusin(function(){

$(this).find("span").css("display","inline").fadeOut(1000);

});

});

</script>

</head>

<body>

<p><input type="text" /> <span>focusout fire</span></p>

<p><input type="password" /> <span>focusout fire</span></p>

</body>

</html>

## **37.3.**blur

当 focusable 元素失去焦点时触发目标函数，不支持冒泡；

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("input").blur(function(){ var val=$(this).val();

alert("Hello 失去焦点:"+val);

});

})

</script>

</head>

<body>

<input type="text" value="刘备" />

<input type="text" value="刘备 2" />

<input type="text" value="刘备 3" />

</body>

</html>

## **37.4.**focusout

和 blur 一样，只是此事件支持冒泡；select

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("p").focusout(function(){

$(this).find("span").css("display","inline").fadeOut(1000);

});

});

</script>

</head>

<body>

<p><input type="text" /> <span>focusout fire</span></p>

<p><input type="password" /> <span>focusout fire</span></p>

</body>

</html>

## **37.5.**change

当下拉框、单选框,文本框的值发生变化时调用

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function sel(val){

console.log(val);

}

</script>

</head>

<body>

<select name="province" onchange="sel(this.value)">

<option value="天津">天津</option>

<option value="河北">河北</option>

<option value="山东">山东</option>

<option value="四川">四川</option>

</select>

</body>

</html>

* + 1. **jquery 的写法**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

/\* function sel(val){

console.log(val);

} \*/

$(function(){

$("#province").change(function(){

//选择哪个省在控制台打印哪个省console.log(this.value);

})

});

</script>

</head>

<body>

<select id="province" name="province"><!-- onchange="sel(this.value)" -->

<option value="天津">天津</option>

<option value="河北">河北</option>

<option value="山东">山东</option>

<option value="四川">四川</option>

</select>

</body>

</html>

* + 1. **文本框的 change,验证**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("#username").change(function(){ alert("用户名验证"+this.value);

//验证用户名:必须是 5~8 位字母数字下划线组成(正则) if(/^\w{5,8}$/.test(this.value)){

alert("验证通过");

}else{

alert("验证没通过:用户名必须是 5~8 位字母数字下划线组成");

}

});

});

</script>

</head>

<body>

用户名:<input type="text" id="username" name="username" /><br/>

密码:<input type="password" id="password" name="password" /><br/>

</body>

</html>

* 1. **submit：**

当表单提交时触发,return false 不提交 return true 提交。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("form").submit(function(){ alert(111);

//执行验证

return false;//验证未通过,不提交

</script>

</head>

<body>

<form action="#" method="get">

<input type="button" id="btn" value="提交"/>

</form>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("#btn").click(function(){

$("form:first").submit();//提交表单

})

})

// return true;//验证通过,提交

});

})

</script>

</head>

<body>

<form action="#" method="get">

<input type="submit" id="btn" value="提交"/>

</form>

</body>

</html>

# **38.**jQuery 事件对象内置函数及属性：

## **38.1.**event.currentTarget

获取绑定事件的 DOM 元素,相当于 this。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("p").click(function(e){

console.log($(this).html()); console.log(e.currentTarget);

console.log(e.currentTarget==this);

})

})

</script>

</head>

<body>

<p>刘备</p>

<p>刘备 2</p>

</body>

</html>

## **38.2.**event.data

传递给执行函数的参数。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("p").click({id:1},function(e){

console.log(e.data.id);//通过 e.data.参数名取得传参的值console.log($(this).html());

console.log(e.currentTarget); console.log(e.currentTarget==this);

})

})

</script>

</head>

<body>

<p>刘备</p>

<p>刘备 2</p>

</body>

</html>

event.isDefaultPrevented()：返回是否调用 event.preventDefault() 方法,来返回一个布尔值。

event.pageX：鼠标相对于文档的左边缘的位置（左边）。event.pageY：鼠标相对于文档的顶部边缘的位置（坐标）。

event.preventDefault()如果调用这个方法，默认事件行为将不再触发。event.target：触发事件的 DOM 元素。

event.timeStamp：这个属性返回事件触发时距离 1970 年 1 月 1 日的毫秒数。

event.type：描述事件的性质

# **39.**jQuery 对象遍历

通过某一个 dom 对象的父子、兄弟之间的关系检索对象

## **39.1.**children(‘selector’)

获得当前元素的全部直接子元素(集合)，可以在参数中使用选择器过滤。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

$("div").children().css("background","red");

}

</script>

</head>

<body>

<p>Hello</p><div><span>Hello Again</span><span>Hello Again</span><span>Hello Again</span><span>Hello Again</span></div><p>And Again</p>

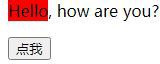
<button type="button" onclick="test()">点我</button>

</body>

</html>

## **39.2.**find(‘selector’)

获得当前元素的全部后代元素，可以在参数中使用选择器过滤。



<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

$("p").find("span").css("background","red");

}

</script>

</head>

<body>

<p><span>Hello</span>, how are you?</p>

<button type="button" onclick="test()">点我</button>

</body>

</html>

## **39.3.**next()

获得当前元素的下一个兄弟元素。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function() {

$("#btn").click(function() {

//找到每个段落的后面紧邻的同辈元素。

$("p").next("p").css("background", "red");

});

});

</script>

</head>

<body>

<p>Hello</p>

<p>Hello Again</p>

<div><span>And Again</span></div>

<button type="button" id="btn">点我</button>

</body>

</html>

## **39.4.**nextAll()

获得当前元素之后的全部兄弟元素。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

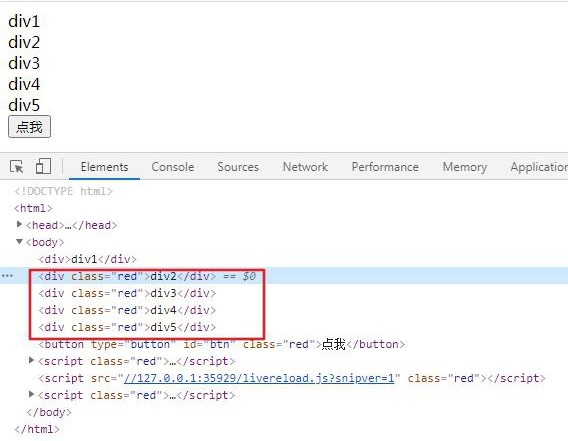
$("#btn").click(function(){

//给第一个 div 之后的所有元素加个类

$("div:first").nextAll().addClass("red");

})

});



</script>

</head>

<body>

<div>div1</div>

<div>div2</div>

<div>div3</div>

<div>div4</div>

<div>div5</div>

<button type="button" id="btn">点我</button>

</body>

</html>

prev():获得当前元素的前一个兄弟元素。prevAll()：获得当前元素之前的全部兄弟元素。

## **39.5.**parent()

获得当前元素的直接父元素。

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("#btn").click(function(){

//查找每个段落的父元素

$("p").parent().css("background","red");

})

})

</script>

</head>

<body>

<div>

<p>Hello</p>

<p>Hello</p>

</div>

<button type="button" id="btn">点我</button>

</body>

</html>

## **39.6.**parents()

获得当前元素的所有父元素。

offsetParent()：获取距离当前元素最近的含有定位信息的祖先元素。含有定位信息的元素指 的是，CSS 的 position 属性是 relative, absolute, 或 fixed 的元素

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript">

$(function(){

$("#btn").click(function(){

//找到每个 span 元素的所有祖先元素。

$("span").parents().css("background","red");

}

)

});

</script>

</head>

<body>

<div>

<p><span>Hello</span></p><span>Hello Again</span>

</div>

<button type="button" id="btn">点我</button>

</body>

</html>

# **40.**jQuery 对于特效的封装都是用动画效果来显示和隐藏元素

基本的隐藏和显示

hide():隐藏匹配的元素。示例： $('img').hide();

show():显示匹配的元素.

示 例 ： $('img').show(); toggle()：显示或隐藏匹配元素。

示例： $('img').toggle();

$('img').toggle(3000,function() {

alert('操作成功');});

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test(){

$("p").hide(3000);

}

function test2(){

$("p").show(2000,function() { alert('操作成功');

})}

</script>

</head>

<body>

<p>

刘备

</p>

<button type="button" onclick="test()">点我</button>

<button type="button" onclick="test2()">点我 2</button>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test1(){

$("img").show(3000);//显示

}

function test2(){

$("img").hide(2000);//隐藏

}

function test3(){

$("img").toggle(1000,function(){ alert("切换完毕")

});//切换

}

</script>

</head>

<body>

<img src="img/1.png" ><br/>

<button type="button" onclick="test1()">显示</button>

<button type="button" onclick="test2()">隐藏</button>

<button type="button" onclick="test3()">切换</button>

</body>

</html>

# **41.**淡入淡出隐藏和显示



<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test1(){

$("img").fadeIn(1000,function(){ alert("淡入完毕");

});

}

function test2(){

$("img").fadeOut(1000,function(){ alert("淡出完毕");

});

}

function test3(){

$("img").fadeTo(1000,0.5,function(){ alert("透明完毕");

});

}

</script>

</head>

<body>

<img src="img/1.png" ><br/>

<button type="button" onclick="test1()">淡入</button>

<button type="button" onclick="test2()">淡出</button>

<button type="button" onclick="test3()">透明度</button>

</body>

</html>

# **42.**滑动的隐藏和显示



<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<script src="js/jquery-1.9.1.js" type="text/javascript" charset="utf-8"></script>

<script type="text/javascript"> function test1(){

$("img").slideDown(1000,function(){ alert("滑动显示完毕");

});

}

function test2(){

$("img").slideUp(1000,function(){ alert("滑动隐藏完毕");

});

}

function test3(){

$("img").slideToggle(1000,function(){ alert("滑动切换完毕");

});

}

</script>

</head>

<body>

<img src="img/1.png" >

<button type="button" onclick="test1()">显示</button>

<button type="button" onclick="test2()">隐藏</button>

<button type="button" onclick="test3()">切换</button>

</body>

</html>