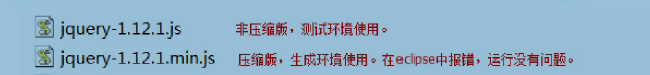
# 一、Jquery入门

## 1、什么是jquery？

Jquery是个js框架，它兼容CSS3，还兼容各种浏览器。

Jquery的核心理念是写的更少，做的更多。

Jquery在2.0以后不再支持IE6/7/8浏览器



## 2、jquery的引入及对象获取

### （1）jquery引入

在线导入：<script src="https://code.jquery.com/jquery-1.8.3.min.js"></script>

在<script>标签中,使用src直接引入jquery的在线路径

项目中导入：<script src="../../../jquery-1.8.3.min.js"></script>

在<script>标签中,使用src直接引入jquery在本地项目中的路径

### （2）jquery语法

（1） jquery(选择器); 或 $(选择器);

（2） 在jquery中, ”**jQuery==$**”

（3）命名约定 jquery变量名建议以$开头

var $name = $(“#username”);

（4）函数写法

jQuery(document).ready(function(){

alert("jQuery(document).ready(function(){});")

});

等同于：

$(function() { //简写

alert("$(function() {});")

});

### （3）jquery与js加载页面的区别

JavaScript 传统页面加载方式存在覆盖问题，加载比jquery慢。当 整个页面，包括里边的内容(比如图片等)全部加载完毕后，才会开始加载JavaScript代码。

jquery的加载比JavaScript快，当dom树结构加载完毕就会加载jquery代码。

jquery不存在覆盖物问题，加载时是顺序加载。

## 3、DOM对象与jQuery对象的转换

1、Dom对象（js对象）无法操作jQuery对象的方法与属性，jQuery对象也无法操作Dom对象的属性与方法。

2、Dom对象与jQuery对象的相互转换

（1）基础语法

**js对象转换成jquery对象**

Var jsObject = document.getElementById(“id”);//js对象

Var $jqObject = $(jsObject);// jQuery对象（js对象转换成jQuery对象）

**jquery对象转换成js对象**

var $jqObject = $(“#id”); // jquery对象

var jsObject = $jqObject.get(0); // jquery对象转换成js对象

var jsObject = $jqObject[0]; // jquery对象转换成js对象

# 二、选择器

## 1、基本选择器

id选择器：$("#id")

类选择器：$(".class")

元素选择器：$("div")

组合选择器：$("#id,.class,div") 多个选择器使用逗号隔开，将所有选中的选择器加入到一个结果中。也就是 result = #id||.class||div

层级选择器：$("#id .class div") 多个选择器使用空格隔开，从第一个选择器开始，一次区下级寻找后边选择器选中的内容。（也就是 在#id 下寻找 .class选择器，再在.class选择器中寻找 div元素选择器。）

## 2、层级选择器

后代选择器：$("body div")，祖先选择器（body），匹配body下的所有后代元素（div）

儿子选择器：$("body>div"),给定的父元素（body），匹配body下的所有子元素（div），只选儿子，不选孙子

弟弟选择器：$("#tow+div"),给定兄弟选择器(#two)，匹配紧挨#two的第一个div元素(#three)

兄弟选择器:$("#one~div"),给定兄弟选择器(#one),匹配后边所有与#one平级的div元素

## 3、基本过滤器

"body div:first"：<body> 中的第一个div元素

"body div:last"：<body> 中的最后一个div元素（子孙辈最后一个）

"body div:odd":<body> 中奇数行的div元素（从0开始，向嵌套中的子孙div数）

"body div:even":<body> 中偶数行的div元素（从0开始，向嵌套中的子孙div数）

## 4、属性选择器

$("body div[id]")：匹配<body> 中带id属性的div元素

$("body div[id='two']")：匹配<body>中 id=two 的div元素

## 5、表单选择器

请自行查阅jquery文档

# 三、jquery的常用方法

## 1、隐藏显示（toggle()）

1.1 案例

需求：隐藏div中的内容

|  |
| --- |
| <script>  // 声明函数  $(function() {  // 动态绑定事件  $("#btn4").click(function() {  // 对图片绑定toggle()效果，toggle()是display的显示和隐藏效果  $("#img4").toggle();  });  });  </script> |

2.2 隔行换色

|  |
| --- |
| $("tbody tr:odd").css("background-color", "pink");  $("tbody tr:even").css("background-color", "green"); |

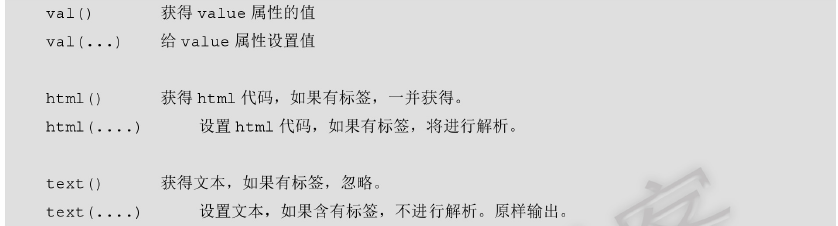
2.3 全选和取消全选

|  |
| --- |
| $(function(){  $("#select").click(function(){  $("tbody input").prop("checked",this.checked);  });  }); |

Jquery1.8.3之前使用attr()方法设置属性，jquery1.8.3之后，使用prop()方法设置属性

# 四、jQuery进阶

## 1、属性操作 val()、 text()、 html()



### 1.1 attr()方法：

attr(name)：根据属性名称获取属性的值

attr(key,value)：设置属性值

attr(properties)：同时设置多个属性

removeAttr(name)：删除某个属性

1.2 注意：

所有方法注意查看运行显示结果，查看其中的某个方法时，请注释其它方法，后面的案例不再说明！

### 2.1 html()方法：

html()方法用于读取innerHTML

html(val)方法用于设置innerHTML

### 3.1 class()方法：

addClass(class) 添加一个class属性

removeClass([class]) 移除一个class属性

toggleClass(class) 如果存在(不存在)就删除(添加)一个类

### 4.1 文本属性操作text()

text()方法用于读取文本内容

text(val)方法用于设置文本内容

### 5.1 值val()操作

val()方法用于读取元素value属性

val(val)方法用于设置元素value属性

## 2、文档处理

### 2.1 插入操作

内部插入

A.append(B):内部结尾处，将B追加到A里面去

A.prepend(B):内部开始处，将B追加到A里面去

A.appendTo(B):内部结尾处，将A追加到B里面去

A.prependTo(B):内部开始处，将B追加到A里面去

外部插入

A.after(B):外部，将B追加到A后面

A.Before(B):外部，将A追加到B前面

A.insertAfter(B):外部，将A追加到B后面

A.insertBefore(B)::外部，将A追加到B前面

注意：

获取下拉框中option的value值的两种方式：

$("option[value='本科']")

$("option:contains('高中')")

## 2.2 克隆操作

clone()：克隆匹配的DOM元素并且选中这些克隆的副本。

clone(true)：元素以及其所有的事件处理并且选中这些克隆的副本(简言之，副本具有与真身一样的事件处理能力)

## 3、替换

A.replaceWith(B); 用B替换掉A的内容

## 4、删除

remove()：删除节点后，事件也会删除

detach()：删除节点后，事件会保留 从1.4新API

empty():清空元素中的所有后代节点。(这个案例是要删除而不是清空，需要注意)

## 5、遍历

### 5.1 使用对象访问方式遍历

语法：$(选择器).each(function(){ });

5.1.1 案例 全选和全不选

|  |
| --- |
| <script type="text/javascript">  $(function() {  // 全选和全不选事件  $("#checkallbox").click(function() {  var isChecked = this.checked; //被选中的状态，返回true或false  // 属性选择器 使用对象访问的方式进行遍历，语法$(选择器).each(function(){ });  $("input[name=hobby]").each(function() {  this.checked = isChecked;//将选择按钮的状态赋值给各个选项  });  });  });  </script> |

### 5.2 使用工具类进行遍历

语法：$.each(array,function(i,j){ });

Array：被遍历的集合

i:集合的脚标

j：被遍历后的内容

5.2.1 案例

|  |
| --- |
| <script type="text/javascript">  $(function() {  $("#checkallbox").click(function() {  var isChecked = this.checked; //当前对象的状态(是否被选中)  // 语法：$.each(array,function(i,j){}); array是被遍历的集合； i是集合的脚标,j是遍历后的内容  $.each($("input[name=hobby]"), function(i, j) {  j.checked = isChecked;  });  });  });  </script> |

## 6、CSS操作

6.1 CSS操作方法汇总：

通过attr属性设置 / 获取 style属性

attr('style', 'color:red'); // 添加style属性

设置CSS样式属性

css(name, value) 设置一个CSS样式属性

css(properties) 传递key - value对象， 设置多个CSS样式属性

设置class属性

addClass(class) 添加一个class属性

removeClass([class]) 移除一个class属性

toggleClass(class) 如果存在(不存在)就删除(添加) 一个类

6.2 案例

|  |
| --- |
| <script>  $(function() {  /\*// 方式一 ：使用css()方法  // 为按钮绑定事件 ---- 背景颜色改为绿色  $("#button1").click(function() {  $("#div1").css("background-color", "green"); //背景色设置为绿色  });  // 为按钮绑定事件 ----背景色改为绿色，字体改为红包  $("#button2").click(function() {  $("#div1").css({  "background-color": "green",  "color": "red"  });  });\*/  // 方式二:使用 attr()方法  $("#button1").click(function() {  $("#div1").attr("style", "background-color:green");  });  $("#button2").click(function() {  $("#div1").css({  "background-color": "green",  "color": "red"  });  });  });  </script> |

## 7、事件绑定 bind

语法：$().bind(name,function(){ });

Name:事件的名称

## 8、事件移除 unbind

语法：$().unbind(name);

# 五、二级省市联动

|  |
| --- |
| <script>  $(function() {  //1.创建二维数组用于存储省份和城市  var cities = new Array(3);  cities[0] = new Array("武汉市", "黄冈市", "襄阳市", "荆州市");  cities[1] = new Array("长沙市", "郴州市", "株洲市", "岳阳市");  cities[2] = new Array("石家庄市", "邯郸市", "廊坊市", "保定市");  cities[3] = new Array("郑州市", "洛阳市", "开封市", "安阳市");  // 2.确定事件 cahnge()  $("#province").change(function() {  // 11.清除二级（城市）元素节点  $("#city").empty();  // 3.获取选择的省份  var valEle = this.value; //  // 4.遍历省份  $.each(cities, function(i, n) {  // 5.判断用户选择都是的省份  if(i == valEle) {  // 6.遍历该省份下的所有城市  $.each(cities[i], function(j, m) {  // 7.创建城市文本节点  var textNode = document.createTextNode(m);  // 8.创建option元素节点  var optionNode = document.createElement("option");  // 9.将文城市本节点添加到option元素节点中去  $(optionNode).append(textNode);  // 10.将元素文本节点追加到第二个下拉列表中去  $(optionNode).appendTo($("#city"));  });  }  });  });  });  </script> |

# 六、下拉列表左右选择

|  |
| --- |
| $(function() {  // 双击将左侧选中放入到右边  $("#left option").dblclick(function() {  $("#left option:selected").appendTo($("#right"));  $.each($("#right option"), function(i, n) {  alert("右边" + i + ":" + n.value);  });  $.each($("#left option"), function(i, n) {  alert("左边" + i + ":" + n.value);  });  });  // 选中单击  $("#addCom").click(function() {  $("#left option:selected").appendTo($("#right"));  alert("addCom");  });  //全部添加到右侧  $("#addComAll").click(function() {  $("#left option").appendTo($("#right"));  alert("addCom");  });  });  // 右侧向左侧移动（删除操作）  $(function() {  // 双击移除商品（右侧添加到左侧）  $("#right option").dblclick(function() {  $("#right option:selected").appendTo($("#left"));  $.each($("#right option"), function(j, m) {  alert(j + ":" + m.value);  });  });  });  </script> |

# 七、表单校验

## 1. 表单校验插件

表单校验插件 validate.min.js

国际化js文件messages\_zh.js

## 2、案例

|  |
| --- |
| <script type="text/javascript" src="../js/jquery-1.8.3.js" ></script>  <!--引入validate插件js文件-->  <script type="text/javascript" src="../js/jquery.validate.min.js" ></script>  <!--引入国际化js文件-->  <script type="text/javascript" src="../js/messages\_zh.js" ></script>  <script>  $(function(){  $("#registForm").validate({  rules:{  user:{  required:true,  minlength:3  },  password:{  required:true,  digits:true,  minlength:6  },  repassword:{  required:true,  equalTo:"[name='password']"  },  email:{  required:true,  email:true  },  username:{  required:true,  maxlength:5  },  sex:{  required:true  }  },  messages:{  user:{  required:"用户名不能为空!",  minlength:"用户名不得少于3位!"  },  password:{  required:"密码不能为空!",  digits:"密码必须是整数!",  minlength:"密码不得少于6位!"  },  repassword:{  required:"确认密码不能为空!",  equalTo:"两次输入密码不一致!"  },  email:{  required:"邮箱不能为空!",  email:"邮箱格式不正确!"  },  username:{  required:"姓名不能为空!",  maxlength:"姓名不得多于5位!"  },  sex:{  required:"性别必须勾选!"  }  },  errorElement: "label", //用来创建错误提示信息标签,validate插件默认的就是label  success: function(label) { //验证成功后的执行的回调函数  //label指向上面那个错误提示信息标签label  label.text(" ") //清空错误提示消息  .addClass("success"); //加上自定义的success类  }  });  });  </script> |

# 八、总结

## 1、jquery入门

### 1.1 引入jquery

<script type="text/javascript" **src="../../../js/jquery-1.8.3.js"**></script>

### 1.2 jquery语法

$(function(){ }); jquery的函数

### 1.3 获取对象

语法：var $id = $(“#id”);

Jquery中，变量的命名常以 $ 开头

### 1.4 jquery对象与Dom对象的转换

Dom对象转jquery对象： var $jqObjectr = $(domObject)；

Jquery对象转Dom对象： var domObject = $jqObject[0];

## 2、jquery基本操作

### 2.1 jquery页面加载

（1） js页面加载

Window.onload(){

Alert(“js页面加载”);

}

（2） jquery页面加载

$(document).ready(){

Alert(“jquery页面加载”);

}

（3）注意：

js给onload只能赋一个值，当赋值多次时，后边的将覆盖前边的

jquery ready可以使用多次，多个页面加载将依次执行

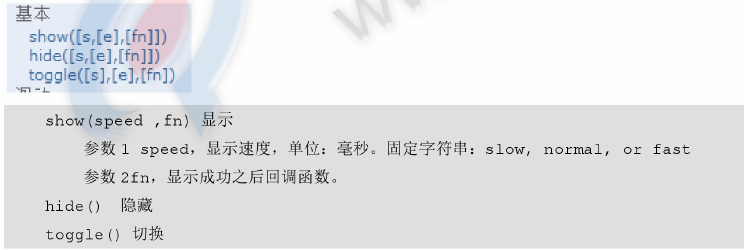
## 3、选择器

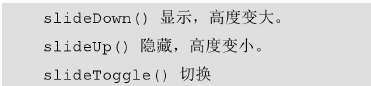
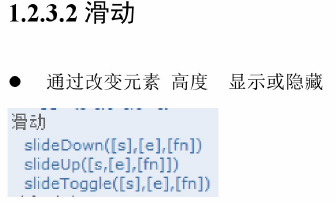
### 3.1 基本选择器

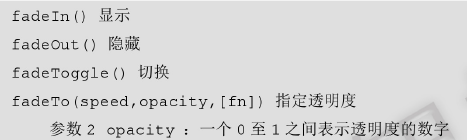
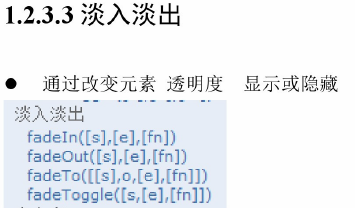
id选择器 格式：$(“#id”);

标签选择器 格式：$(“input”)

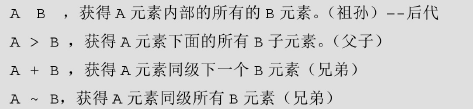
类选择器 格式：$(“.class”);



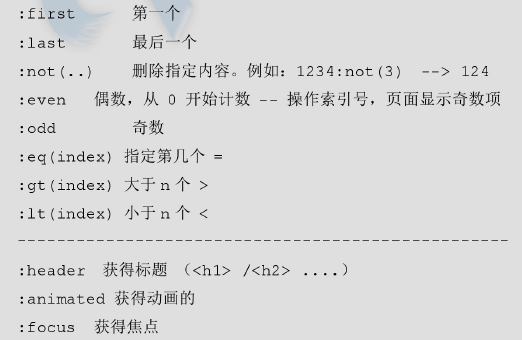




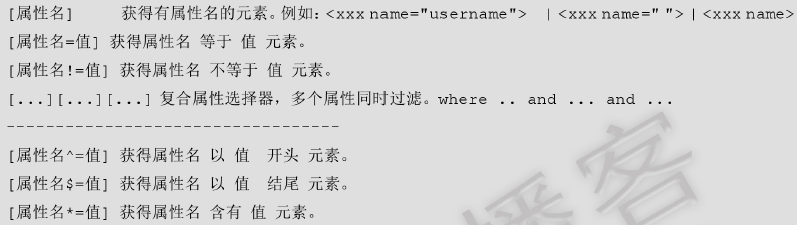
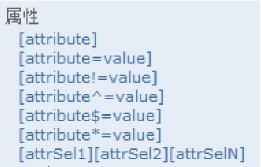
### 3.2 层级选择器



### 3.3 基本过滤选择器



### 3.4 属性选择器

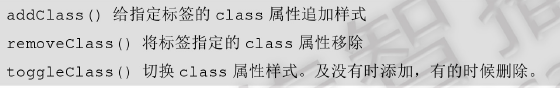


### 3.5表单属性选择器



## 4-1、换色

### 4.1 相关技术

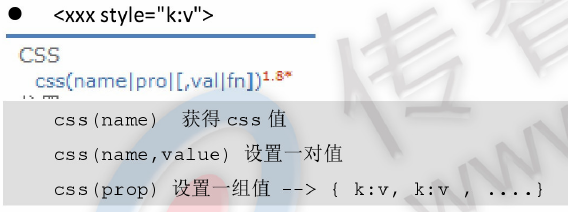


### 4.2 案例实现

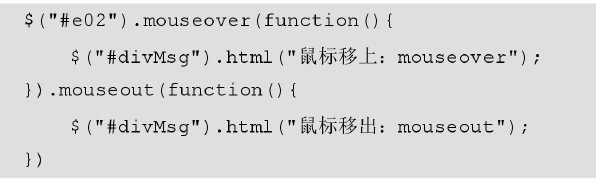


## 4-2、换色另一种实现方式

### 4.1 CSS样式操作



### 4.2 鼠标事件



### 4.3 案例实现



## 5、全选和全不选

### 5.1 属性操作 pop()

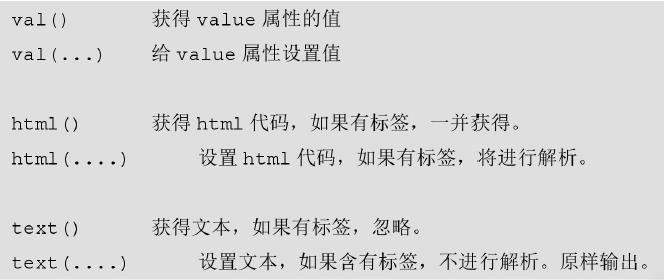
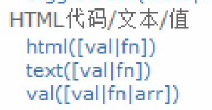


### 5.2 案例实现

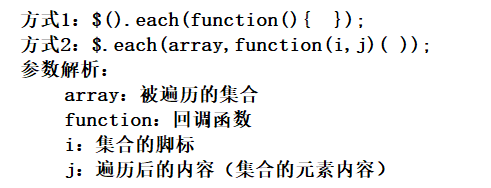


## 6、省市联动

### 6.1 属性操作

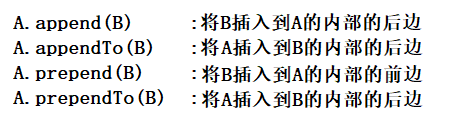


### 6.2 遍历函数：each

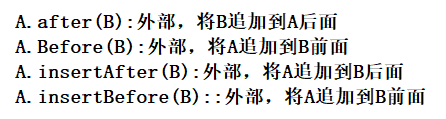


### 6.3 文档处理

6.3.1 内部插入

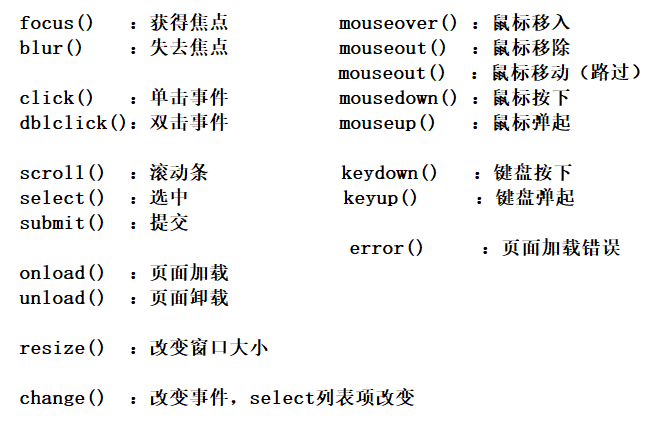


6.3.2 外部插入

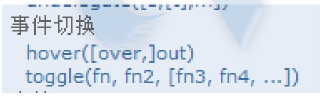
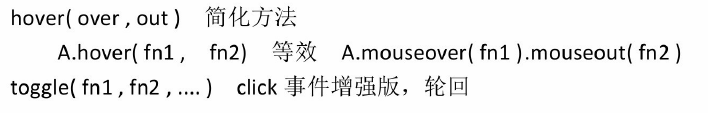


## 7、事件

### 7.1 常用事件



### 7.2 事件切换

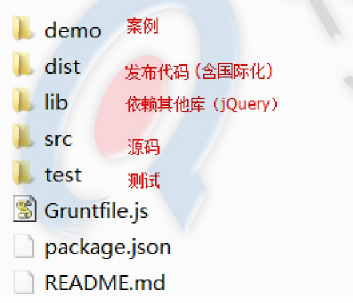
 

## 8、表单校验

### 8.1 表单插件 validation

8.1.1 目录结构

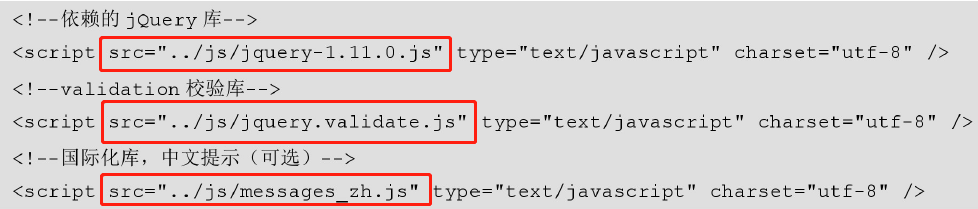




### 8.2 导入

Validation 插件是在jquery的基础上运行的，我们分三步导入：

1. Jquery 框架脚本
2. Validation 插件脚本
3. Messages\_zh 国际化脚本



8.3 校验规则



8.4 语法