jquery基础教程:JQuery 常用思路方法基础教程

疯狂代码 http://www.crazycoder.cn/ ĵ:http:/www.crazycoder.cn/Javascript/Article62552.html

$(“p”).addClass(css中定义样式类型); 给某个元素添加样式

$(“img”).attr({src:”test.jpg”,alt:”test Image”}); 给某个元素添加属性/值参数是map

$(“img”).attr(“src”,”test.jpg”); 给某个元素添加属性/值

$(“img”).attr(“title”, function { this.src }); 给某个元素添加属性/值

$(“元素名称”).html; 获得该元素内内容(元素文本等)

$(“元素名称”).html(“ stuff”); 给某元素设置内容

$(“元素名称”).removeAttr(“属性名称”) 给某元素删除指定属性以及该属性值

$(“元素名称”).removeClass(“”) 给某元素删除指定样式

$(“元素名称”).text; 获得该元素文本

$(“元素名称”).text(value); 设置该元素文本值为value

$(“元素名称”).toggleClass() 当元素存在参数中样式时候取消,如果不存在就设置此样式

$(“input元素名称”).val; 获取input元素值

$(“input元素名称”).val(value); 设置input元素值为value

Manipulation:

$(“元素名称”).after(content); 在匹配元素后面添加内容

$(“元素名称”).append(content); 将content作为元素内容插入到该元素后面

$(“元素名称”).appendTo(content); 在content后接元素

$(“元素名称”).before(content); 和after思路方法相反

$(“元素名称”).clone(布尔表达式) 当布尔表达式为真时克隆元素(无参时当作true处理)

$(“元素名称”).empty 将该元素内容设置为空

$(“元素名称”).insertAfter(content); 将该元素插入到content的后

$(“元素名称”).insertBefore(content); 将该元素插入到content的前

$(“元素”).prepend(content); 将content作为该元素部分放到该元素最前面

$(“元素”).prependTo(content); 将该元素作为content部分放content最前面

$(“元素”).remove; 删除所有指定元素

$(“元素”).remove(“exp”); 删除所有含有exp元素

$(“元素”).wrap(“html”); 用html来包围该元素

$(“元素”).wrap(element); 用element来包围该元素

Traversing:

add(expr)

add(html)

add(elements)

children(expr)

contains(str)
end
filter(expression)
filter(filter)
find(expr)
is(expr)
next(expr)
not(el)
not(expr)
not(elems)
parent(expr)
parents(expr)
prev(expr)
siblings(expr)
Core:
$(html).appendTo(" body ") 相当于在body中写了段html代码
$(elems) 获得DOM上某个元素
$(function{……..}); 执行个
$(" div > p ").css(" border ", "1px solid gray "); 查找所有div子节点p添加样式
$(" input:radio ", document.forms[0]) 在当前页面第个表单中查找所有单选按钮
$.extend(prop) prop是个jquery对象
举例:
jQuery.extend({
min: function(a, b) { a < b ? a : b; },
max: function(a, b) { a > b ? a : b; }
});
jQuery(expression, [context]) —$(expression, [context]); 在默认情况下$查询是当前HTML文档中DOM元素
each(callback) 以每个匹配元素作为上下文来执行个
举例:1
$(" span ").click(function){
$(" li ").each(function{
$(this).toggleClass(" example ");
});
});
举例:2

```
$(" button").click(function {
$(" div").each(function (index, domEle) {
// domEle this
$(domEle).css(" backgroundColor", "yellow");
($(this).is(" #stop")) {
$(" span").text(" Stopped at div index #" + index);
false;
}
});
});
```

jQuery Event:
ready(fn); $(document).ready注意在body中没有onload事件否则该不能执行在每个页面中可以有很多个被加载执行按照fn顺序来执行
bind(type, [data], fn) 为每个匹配元素特定事件(像click)绑定个或多个事件处理器可能事件属性有:blur, focus, load, resize, scroll, unload, click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, change, select, submit, keydown, keypress, keyup, error
one(type, [data], fn) 为每个匹配元素特定事件(像click)绑定个或多个事件处理器在每个对象上这个事件处理只会被执行次其他规则和bind相同
trigger(type, [data]) 在每个匹配元素上触发某类事件
triggerHandler(type, [data]) 这特定思路方法会触发个元素上特定事件(指定个事件类型)同时取消浏览器对此事件默认行动
unbind([type], [data]) 反绑定从每个匹配元素中删除绑定事件
$(" p").unbind 移除所有段落上所有绑定事件
$(" p").unbind("click") 移除所有段落上click事件
hover(over, out) over,out都是思路方法, 当鼠标移动到个匹配元素上面时会触发指定第个当鼠标移出这个元素时会触发指定第 2个

```
$(" p").hover(function{
$(this).addClass(" over");
},
function{
$(this).addClass(" out");
}
);
```

toggle(fn, fn) 如果点击了个匹配元素则触发指定第个当再次点击同元素时则触发指定第 2个

$(“ p”).toggle(function{
$(this).addClass(“ selected”);
},
function{
$(this).removeClass(“ selected”);
}
);
元素事件列表介绍说明
注:不带参数其参数为可选 fnjQuery不支持form元素re事件
事件 描述 支持元素或对象
blur( ) 元素失去焦点 a, input, textarea, button, select, label, map, area
change( ) 用户改变域内容 input, textarea, select
click( ) 鼠标点击某个对象 几乎所有元素
dblclick( ) 鼠标双击某个对象 几乎所有元素
error( ) 当加载文档或图像时发生某个 window, img
focus( ) 元素获得焦点 a, input, textarea, button, select, label, map, area
keydown( ) 某个键盘键被按下 几乎所有元素
keypress( ) 某个键盘键被按下或按住 几乎所有元素
keyup( ) 某个键盘键被松开 几乎所有元素
load(fn) 某个页面或图像被完成加载 window, img
mousedown(fn) 某个鼠标按键被按下 几乎所有元素
mousemove(fn) 鼠标被移动 几乎所有元素
mouseout(fn) 鼠标从某元素移开 几乎所有元素
mouseover(fn) 鼠标被移到某元素的上 几乎所有元素
mouseup(fn) 某个鼠标按键被松开 几乎所有元素
resize(fn) 窗口或框架被调整尺寸 window, rame, frame
scroll(fn) 滚动文档可视部分时 window
select( ) 文本被选定 document, input, textarea
submit( ) 提交按钮被点击 form
unload(fn) 用户退出页面 window
JQuery Ajax 思路方法介绍说明:
load(url, [data], [callback]) 装入个远程HTML内容到个DOM结点
$(“ #feeds”).load(“ feeds.html”); 将feeds.html文件载入到id为feedsdiv中
$(“ #feeds”).load(“ feeds.php”, {limit: 25}, function{
alert(“ The last 25 entries in the feed have been loaded”);
});

jQuery.get(url, [data], [callback]) 使用GET请求个页面
$.get(“ test.cgi”, { name: “John”, time: “2pm” }, function(data){
alert(“ Data Loaded: ” + data);
});
jQuery.getJSON(url, [data], [callback]) 使用GET请求JSON数据
$.getJSON(“ test.js”, { name: “John”, time: “2pm” }, function(json){
alert(“ JSON Data: ” + json.users[3].name);
});
jQuery.getScript(url, [callback]) 使用GET请求javascript文件并执行
$.getScript(“ test.js”, function{
alert(“ Script loaded and executed.”);
});
jQuery.post(url, [data], [callback], [type]) 使用POST请求个页面
ajaxComplete(callback) 当个AJAX请求结束后执行个这是个Ajax事件
$(“ #msg”).ajaxComplete(function(request, tings){
$(this).append(“ Request Complete.”);
});
ajaxError(callback) 当个AJAX请求失败后执行个这是个Ajax事件
$(“ #msg”).ajaxError(function(request, tings){
$(this).append(“ Error requesting page ” + tings.url + “”);
});
ajaxSend(callback) 在个AJAX请求发送时执行个这是个Ajax事件
$(“ #msg”).ajaxSend(function(evt, request, tings){
$(this).append(“ <li<Starting request at ” + tings.url
+ “</li<”);
});
ajaxStart(callback) 在个AJAX请求开始但还没有激活时执行个这是个Ajax事件
当AJAX请求开始(并还没有激活时)显示loading信息
$(“ #loading”).ajaxStart(function{
$(this).show;
});
ajaxStop(callback) 当所有AJAX都停止时执行个这是个Ajax事件
当所有AJAX请求都停止时隐藏loading信息
$(“ #loading”).ajaxStop(function{
$(this).hide;
});

ajaxSuccess(callback) 当个AJAX请求成功完成后执行个这是个Ajax事件
当AJAX请求成功完成时显示信息
$(“ #msg”).ajaxSuccess(function(evt, request, tings){
$(this).append(“ Successful Request!”);
});
jQuery.ajaxSetup(options) 为所有AJAX请求进行全局设置查看$.ajax取得所有选项信息
设置默认全局AJAX请求选项
$.ajaxSetup({
url: “/xmlhttp/”,
global: false,
type: “POST”
});
$.ajax({ data: myData });
serialize( ) 以名称和值方式连接组input元素实现了正确表单元素序列
function showValues {
var str = $(“ form”).serialize;
$(“ #results”).text(str);
}
$(“ :checkbox, :radio”).click(showValues);
$(“ select”).change(showValues);
showValues;
serializeArray( ) 连接所有表单和表单元素(类似于.serialize思路方法)但是返回个JSON数据格式
从form中取得组值显示出来
function showValues {
var fields = $(“ :input”).serializeArray;
alert(fields);
$(“ #results”).empty;
jQuery.each(fields, function(i, field){
$(“ #results”).append(field.value + “ “);
});
}
$(“ :checkbox, :radio”).click(showValues);
$(“ select”).change(showValues);
showValues;
JQuery Effects 思路方法介绍说明
show( ) 显示隐藏匹配元素

show(speed, [callback]) 以优雅动画显示所有匹配元素并在显示完成后可选地触发个回调
hide( ) 隐藏所有匹配元素
hide(speed, [callback]) 以优雅动画隐藏所有匹配元素并在显示完成后可选地触发个回调
toggle( ) 切换元素可见状态如果元素是可见切换为隐藏；如果元素是隐藏
切换为可见
slideDown(speed, [callback]) 通过高度变化(向下增大)来动态地显示所有匹配元素在显示完成后可选
地触发个回调这个动画效果只调整元素高度可以使匹配元素以
“滑动”方式显示出来
slideUp(speed, [callback]) 通过高度变化(向上减小)来动态地隐藏所有匹配元素在隐藏完成后可选地
触发个回调这个动画效果只调整元素高度可以使匹配元素以”滑动”
方式隐藏起来
slideToggle(speed, [callback]) 通过高度变化来切换所有匹配元素可见性并在切换完成后可选地触发个回
调 这个动画效果只调整元素高度可以使匹配元素以”滑动”方式隐
藏或显示
fadeIn(speed, [callback]) 通过不透明度变化来实现所有匹配元素淡入效果并在动画完成后可选地触
发个回调 这个动画只调整元素不透明度也就是说所有匹配元素
高度和宽度不会发生变化
fadeOut(speed, [callback]) 通过不透明度变化来实现所有匹配元素淡出效果并在动画完成后可选地触
发个回调 这个动画只调整元素不透明度也就是说所有匹配元素
高度和宽度不会发生变化
fadeTo(speed, opacity, [callback]) 把所有匹配元素不透明度以渐进方式调整到指定不透明度并在动画完成
后可选地触发个回调 这个动画只调整元素不透明度也就是说所
有匹配元素高度和宽度不会发生变化
stop( ) 停止所有匹配元素当前正在运行动画如果有动画处于队列当中他们就会立即开始
queue( ) 取得第个匹配元素动画序列引用(返回个内容为)
queue(callback) 在每个匹配元素事件序列末尾添加个可执行作为此元素事件
queue(queue) 以个新动画序列代替所有匹配元素原动画序列
dequeue( ) 执行并移除动画序列前端动画
animate(params, [duration], [easing], [callback]) 用于创建自定义动画
animate(params, options) 创建自定义动画另个思路方法作用同上
JQuery Traversing 思路方法介绍说明
eq(index) 从匹配元素集合中取得个指定位置元素index从0开始
filter(expr) 返回和指定表达式匹配元素集合可以使用”,”号分割多个expr用于实现多个条件筛选
filter(fn) 利用个特殊来作为筛选条件移除集合中不匹配元素
is(expr) 用个表达式来检查当前选择元素集合如果其中至少有个元素符合这个给定
表达式就返回true

map(callback) 将jQuery对象中组元素利用callback思路方法转换其值然后添加到个jQuery中
not(expr) 从匹配元素集合中删除和指定表达式匹配元素
slice(start, [end]) 从匹配元素集合中取得个子集和内建slice思路方法相同
add(expr) 把和表达式匹配元素添加到jQuery对象中
children([expr]) 取得个包含匹配元素集合中每个元素所有子元素元素集合可选过滤器
将使这个思路方法只匹配符合元素(只包括元素节点不包括文本节点)
contents( ) 取得个包含匹配元素集合中每个元素所有子孙节点集合(只包括元素节点不
包括文本节点)如果元素为rame则取得其中文档元素
find(expr) 搜索所有和指定表达式匹配元素
next([expr]) 取得个包含匹配元素集合中每个元素紧邻后面同辈元素元素集合
nextAll([expr]) 取得个包含匹配元素集合中每个元素所有后面同辈元素元素集合
parent([expr]) 取得个包含着所有匹配元素唯父元素元素集合
parents([expr]) 取得个包含着所有匹配元素唯祖先元素元素集合(不包含根元素)
prev([expr]) 取得个包含匹配元素集合中每个元素紧邻前个同辈元素元素集合
prevAll([expr]) 取得个包含匹配元素集合中每个元素的前所有同辈元素元素集合
siblings([expr]) 取得个包含匹配元素集合中每个元素所有同辈元素元素集合
andSelf( ) 将前个匹配元素集合添加到当前集合中
取得所有div元素和其中p元素添加border类属性取得所有div元素中p元素
添加background类属性
$(“ div”).find(“ p”).andSelf.addClass(“ border”);
$(“ div”).find(“ p”).addClass(“ background”);
end( ) 结束当前操作回到当前操作前个操作
找到所有p元素其中span元素集合然后返回p元素集合添加css属性
$(“ p”).find(“ span”).end.css(“ border”, “2px red solid”);
JQuery Selectors 思路方法介绍说明
基本选择器
$(“ #myDiv”）匹配唯具有此id值元素
$(“ div”）匹配指定名称所有元素
$(“ .myClass”）匹配具有此样式值所有元素
$(“ *”）匹配所有元素
$(“ div,span,p.myClass”）联合所有匹配选择器
层叠选择器
$(“ form input”）后代选择器选择ancestor所有子孙节点
$(“ # > *”）子选择器选择parent所有子节点
$(“ label + input”）临选择器选择prev下个临节点
$(“ #prev ~ div”）同胞选择器选择prev所有同胞节点

基本过滤选择器
$(“ tr:first”）匹配第个选择元素
$(“ tr:last”）匹配最后个选择元素
$(“ input:not(:checked) + span”)从原元素集合中过滤掉匹配selector所有元素(这里有是个临选择器)
$(“ tr:even”）匹配集合中偶数位置所有元素(从0开始)
$(“ tr:odd”）匹配集合中奇数位置所有元素(从0开始)
$(“ td:eq(2)”）匹配集合中指定位置元素(从0开始)
$(“ td:gt(4)”）匹配集合中指定位置的后所有元素(从0开始)
$(“ td:gl(4)”）匹配集合中指定位置的前所有元素(从0开始)
$(“ :header”）匹配所有标题
$(“ div:animated”）匹配所有正在运行动画所有元素
内容过滤选择器
$(“ div:contains('John')”）匹配含有指定文本所有元素
$(“ td:empty”）匹配所有空元素(只含有文本元素不算空元素)
$(“ div:has(p)”）从原元素集合中再次匹配所有至少含有个selector所有元素
$(“ td:parent”）匹配所有不为空元素(含有文本元素也算)
$(“ div:hidden”）匹配所有隐藏元素也包括表单隐藏域
$(“ div:visible”）匹配所有可见元素
属性过滤选择器
$(“ div[id]”）匹配所有具有指定属性元素
$(“ input[name='sletter']”）匹配所有具有指定属性值元素
$(“ input[name!='sletter']”）匹配所有不具有指定属性值元素
$(“ input[name^='s']”）匹配所有指定属性值以value开头元素
$(“ input[name$='letter']”）匹配所有指定属性值以value结尾元素
$(“ input[name*='man']”）匹配所有指定属性值含有value元素
$(“ input[id][name$='man']”）匹配同时符合多个选择器所有元素
子元素过滤选择器
$(“ ul li:nth-child(2)”),
$(“ ul li:nth-child(odd)”), 匹配父元素第n个子元素
$(“ ul li:nth-child(3n + 1)”）
$(“ div span:first-child”）匹配父元素第1个子元素
$(“ div span:last-child”）匹配父元素最后1个子元素
$(“ div button:only-child”）匹配父元素唯1个子元素
表单元素选择器
$(“ :input”）匹配所有表单输入元素包括所有类型input, textarea, select 和 button
$(“ :text”）匹配所有类型为textinput元素

$(“:password”）匹配所有类型为passwordinput元素
$(“:radio”）匹配所有类型为radioinput元素
$(“:checkbox”）匹配所有类型为checkboxinput元素
$(“:submit”）匹配所有类型为submitinput元素
$(“:image”）匹配所有类型为imageinput元素
$(“:re”）匹配所有类型为reinput元素
$(“:button”）匹配所有类型为buttoninput元素
$(“:file”）匹配所有类型为fileinput元素
$(“:hidden”）匹配所有类型为hiddeninput元素或表单隐藏域
表单元素过滤选择器
$(“:enabled”）匹配所有可操作表单元素
$(“:disabled”）匹配所有不可操作表单元素
$(“:checked”）匹配所有已点选元素
$(“select option:selected”）匹配所有已选择元素
JQuery CSS 思路方法介绍说明
css(name) 访问第个匹配元素样式属性
css(properties) 把个”名/值对”对象设置为所有匹配元素样式属性
$(“p”).hover(function {
$(this).css({ backgroundColor:”yellow”, fontWeight:”bolder” });
}, function {
var cssObj = {
backgroundColor: “#ddd”,
fontWeight: “”,
color: “rgb(0,40,244)”
}
$(this).css(cssObj);
});
css(name, value) 在所有匹配元素中设置个样式属性值
off( ) 取得匹配第个元素相对于当前可视窗口位置返回对象有2个属性
top和left属性值为整数这个只能用于可见元素
var p = $(“p:last”);
var off = p.off;
p.html(“left:” + off.left + “, top:” + off.top);
width( ) 取得当前第匹配元素宽度值
width(val) 为每个匹配元素设置指定宽度值
height( ) 取得当前第匹配元素高度值

height(val) 为每个匹配元素设置指定高度值
JQuery Utilities 思路方法介绍说明
jQuery.browser
.msie 表示ie
jQuery.browser.version 读取用户浏览器版本信息
jQuery.boxModel 检测用户浏览器针对当前页显示是否基于w3c CSS盒模型
jQuery.isFunction(obj) 检测传递参数是否为function
function stub { }
var objs = [
function {},
{ x:15, y:20 },
null,
stub,
"function"
];
jQuery.each(objs, function (i) {
var isFunc = jQuery.isFunction(objs[i]);
$(" span:eq(" + i + ")").text(isFunc);
});
jQuery.trim(str) 清除串两端空格使用正则表达式来清除给定两端空格
jQuery.each(object, callback) 个通用迭代器可以用来无缝迭代对象和
jQuery.extend(target, object1, [objectN]) 扩展个对象修改原来对象并返回这是个强大实现继承
工具这种继承是采用传值思路方法来实现而不是JavaScript中
原型链方式
合并tings和options对象返回修改后tings对象
var tings = { validate: false, limit: 5, name: "foo" };
var options = { validate: true, name: "bar" };
jQuery.extend(tings, options);
合并defaults和options对象defaults对象并没有被修改options对象中值
代替了defaults对象值传递给了empty
var empty = {}
var defaults = { validate: false, limit: 5, name: "foo" };
var options = { validate: true, name: "bar" };
var tings = $.extend(empty, defaults, options);
jQuery.grep(.gif' />, callback, [invert]) 通过个筛选来去除中项
$.grep([0,1,2], function(n,i){

n > 0;
});
jQuery.makeArray(obj) 将个类似对象转化为个真正
将选取div元素集合转化为个
var arr = jQuery.makeArray(document.getElementsByTagName(“ div”));
arr.reverse; // use an Array method on list of dom elements
$(arr).appendTo(document.body);
jQuery.map(.gif' />, callback) 使用某个思路方法修改个中项然后返回个新
jQuery.inArray(value, .gif' />) 返回value在中位置如果没有找到则返回-1
jQuery.unique(.gif' />) 删除中所有重复元素返回整理后
2009-2-10 11:35:43
疯狂代码 http://www.crazycoder.cn/