

前端

基础知识

Get内容介绍

下面的内容，就是GET请求的HTTP协议内容

请求行:

1. 请求的方式 GET
2. 请求的资源路径 /06_servlet/a.html
3. 请求的协议的版本号 HTTP/1.1

```
GET /06_servlet/a.html HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, */*
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SL002; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: localhost:8080
Connection: Keep-Alive
```

请求头

Accept: 告诉服务器，客户端可以接收的数据类型
Accept-Language: 告诉服务器客户端可以接收的语言类型
zh-CN 中文中国
en-US 英文美国
User-Agent: 就是浏览器的信息
Accept-Encoding: 告诉服务器，客户端可以接收的数据编码（压缩）格式
Host: 表示请求的服务器ip和端口号
Connection: 告诉服务器请求连接如何处理
Keep-Alive 告诉服务器回传数据不要马上关闭，保持一小段时间的连接
Closed 马上关闭

Post内容介绍

以下是POST请求HTTP协议内容

请求行:

1. 请求的方式 POST
2. 请求的资源路径 /06_servlet/hello3
3. 请求的协议和版本号 HTTP/1.1

```
POST /06_servlet/hello3 HTTP/1.1
Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, */*
Referer: http://localhost:8080/06_servlet/a.html
Accept-Language: zh-CN
User-Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SL002; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0)
Content-Type: application/x-www-form-urlencoded
UA-CPU: AMD64
Accept-Encoding: gzip, deflate
Host: localhost:8080
Content-Length: 26
Connection: Keep-Alive
Cache-Control: no-cache
```

请求体，发送给服务器的数据

action:login&username=root

请求头

Accept: 表示客户端可以接收的数据类型
Accept-Language: 表示客户端可以接收的语言类型
Referer: 表示请求发起时，浏览器地址栏中的地址（从哪来）
User-Agent: 表示浏览器的信息
Content-Type: 表示发送的数据的类型
application/x-www-form-urlencoded
表示提交的数据格式是: name=value&name=value, 然后对其进行url编码
url编码是把非英文内容转换为: %xx%xx
multipart/form-data
Content-Length: 表示发送的数据的长度
Cache-Control: 表示如何控制缓存 no-cache不缓存

那些是get请求

GET 请求有哪些：

- 1、form 标签 method=get
- 2、a 标签
- 3、link 标签引入 css
- 4、Script 标签引入 js 文件
- 5、img 标签引入图片
- 6、iframe 引入 html 页面
- 7、在浏览器地址栏中输入地址后敲回车

那些是post请求

POST 请求有哪些：

- 8、form 标签 method=post

服务器接受响应

以下内容就是响应的HTTP协议示例：

```
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Accept-Ranges: bytes
ETag: W/"375-1578903773217"
Last-Modified: Mon, 13 Jan 2020 08:22:53 GMT
Content-Type: text/html
Content-Length: 375
Date: Mon, 13 Jan 2020 08:38:42 GMT

<!DOCTYPE html>
<html lang="zh_CN">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
</head>
<body>
  <form action="http://localhost:8080/06_servlet/hello3" method="post">
    <input type="hidden" name="action" value="login" />
    <input type="hidden" name="username" value="root" />
    <input type="submit">
  </form>
</body>
</html>
```

响应行

1、响应的协议
2、响应状态码
3、响应状态描述符

HTTP/1.1
200
OK

响应头

Server: 表示服务器的信息
Content-Type: 表示响应体的数据类型
Content-Length: 响应体的长度
Date: 请求响应的时间 (这是格林时间)

空行

响应体

常见的响应码

- | | |
|-----|----------------------------------|
| 200 | 表示请求成功 |
| 302 | 表示请求重定向 (明天讲) |
| 404 | 表示请求服务器已经收到了,但是你要的数据不存在 (请求地址错误) |
| 500 | 表示服务器已经收到请求,但是服务器内部错误 (代码错误) |

405 请求是get,但是java程序是post

HTTP Status 500 - Unable to compile class for JSP:

type Exception report

message Unable to compile class for JSP:

description The server encountered an internal error that prevented it from fulfilling this request.

exception

org.apache.jasper.JasperException: Unable to compile class for JSP:

An error occurred at line: 10 in the jsp file: /pages/user/./common/head_infomation.jsp
Duplicate field regist_jsp.aaa

```
7: --%>
8: <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9: <%!
10:     String aaa = request.getScheme()
11:         + "://"
12:         + request.getServerName()
13:         + ":"
```

An error occurred at line: 10 in the jsp file: /pages/user/./common/head_infomation.jsp
request cannot be resolved

```
7: --%>
8: <%@ page contentType="text/html; charset=UTF-8" language="java" %>
9: <%!
10:     String aaa = request.getScheme()
11:         + "://"
12:         + request.getServerName()
13:         + ":"
```

引用了2个 include 指向相同的jsp网页

发送文件时HTTP协议内容

HTML

font(字体标签)

//设置字体为红色,宋体,7号大小(1-7)

我是字体标签

特殊字符

空格
< 小于号 <
> 大于号 >
& 和号 &
" 引号 "
' 撇号 '
¢ 分 ¢
£ 镑 £
¥ 日圆 ¥
§ 节 §
© 版权 ©
® 注册商标 ®
× 乘号 ×
÷ 除号 ÷

h1-h6

```
<h1>马佳盛</h1>
```

超链接

```
//当前页面跳转
<a href="跳转网址" target="_self">马佳盛</a>
//打开新页面显示
<a href="跳转网址" target="_blank">马佳盛</a>
```

列表

无序列表

```
<ul>
    <li>马佳盛1</li> //结果 ●值1
    <li>马佳盛2</li> //结果 ●值2
</ul> 想去掉●就在ul后面加上type=None
```

有序列表

```
<ol>
    <li>马佳盛</li> //结果 1.马佳盛
    <li>马佳盛</li> //结果 1.马佳盛
</ol>
```

图片img

```

```

表格

简单的表格

table标签:

border="边框宽多少" align="靠哪边对齐" width="宽多少px" height="高多少px" cellpadding="边框边距"

tr:

全默认

th:

加粗并居中对齐

```
<table border="1" align="center" width="300px" height="300px"
cellspacing="0">
    <tr>
        <td>1.1</td>
        <td>1.2</td>
    </tr>
    <tr>
        <th>1.1</th>
        <th>1.2</th>
    </tr>
</table>
```

跨行跨列表格

在td或者th后面追加属性 colspan="左跨几行" rowspan="右跨几行"

iframe(小页面显示)

说明:在页面中开辟一块空间显示其他网页,一般搭配a标签使用

步骤:在iframe中设置name属性,然后再a标签的target属性设置iframename的值

```
<iframe src="显示的网页" width="宽px" height="高px" name="aaa">
<a href="跳转的网页" target="aaa">跳转</a>
```

表单

表单提交数据不全的问题

- 1.input各个标签没有name属性
- 2.单选、复选、select(option) 都要加上value属性,value="发送给服务器的值"
- 3.input标签没有在form中

Get和Post的区别

Get: 发送字符串为:

- 1.服务器地址:name = value & password & value
- 2.不安全,信息会显示出来
- 3.长度限制为100个字符

Post: 发送字符串为:

- 1.服务器地址,其他说明都没有
- 2.相对于Get安全
- 3.长度没有限制

基础元素

text文本框:

value默认值 maxlength最大长度

password密码框:

自动隐藏值 value默认值 maxlength最大长度

radio单选框:

name进行分类 checked默认选项

checkbox多选框:

checked默认选项

select下拉框:

option下拉框选项 selected默认值

多行多列文本框:

cols宽 rows高 <>中键输入默认值</>

Form表单:

action提交服务器地址

method="提交方式(get|post)" 默认值是get

```
<form action="192.168.1.1" method="post">
```

用户名称:

```
<input type="text" value="默认值" maxlength="10"><br>
```

用户密码:

```
<input type="password" value="默认值"maxlength="10"><br>
```

确认密码:

```
<input type="password" value="默认值"maxlength="10"><br>
```

性别:

```
<input type="radio" name="sex" checked="checked">男 <input  
type="radio" name="sex">女<br>
```

兴趣爱好:

```
<input type="checkbox" checked="checked">java  
<input type="checkbox">javascript
```

国籍: <select>

```
    <option value="">请选择国籍</option>  
    <option selected="selected">中国</option>
```

```
        <option >小日本鬼子</option>
        <option>美国</option>
    </select>

    自我评价：
    <textarea name="" id="" cols="30" rows="10">中间输入默认值</textarea>
    <br>

    reset重置按钮：
    <input type="reset" value="重置啊">

    submit提交按钮：
    <input type="submit" value="我要提交了">

    普通按钮：
    <input type="button" value="按钮">

    file上传文件：
    <input type="file" name="上传文件" id="">

    hidden隐藏区域：
    <input type="hidden" name="">隐藏域
</form>
```

格式化表单

CSS

常用的样式

```
//背景颜色
background-color: red;

//颜色
color:black

//相对定位
position:relative; top left right bottom

//绝对定位
position: absolute; top left right bottom

//宽
width:88px;

//高
height:88px;

//字体高度
```

```
line-height:88px;

//文字位置
text-align:center;

//背景图片,不平铺
background: url(../images/yes_15.png) no-repeat;

//右边距为
margin-right: 88px;

//左边距
margin-left:88px;

//上边距
margin-top:88px;

//下边距
margin-bottom:88px;

//边框,边框宽度,什么线条,什么颜色
border: 0.02rem solid #d7d7d7;

//边框变圆多少0-1
border-radius: 0.5;

//变为块内元素
display: inline-block;

//变味块元素
display: block;

//字体大小
font-size: 32px;

//什么字体
font-family: PingFangSC-Medium,sans-serif;

//列表不显示编号
list-style: none;

//溢出隐藏
overflow:hidden;

//文字溢出隐藏
text-overflow: ellipsis;

//浮动
float: left;
```


JavaScript

加载方式

1.内嵌式

```
//在body中添加script标签,然后再script中写入代码  
<script type="text/javascript"> 语句 </script>
```

2.外链式

```
//在head中添加script标签,src标签位置  
<script src="script文件路径"></script>
```

输出语句

```
alert("hello world11111"); //输出弹框  
console.log("hello world"); //在console中输出hello world document.write("  
<b>doncument</b>"); //输出到网页上 可以加一些标签
```

变量

分类

数值类型 number
字符串类型 string
对象类型 object
布尔类型 boolean
函数类型 function
特殊的值 undefined
未定义,没有赋值的变量默认都是undefined null
空值 NAN
非数字,非数值 0、null、undefined 都是false

定义方式

```
var 变量名 = 值; typeof(变量名) 返回变量的类型
```

运算符

等于: == java一样
全等于: === 不仅比较值,还比较类型
且运算:
1.表达式都为真的情况下返回最后一个表达式的值
2.表达式为假的时候返回第一个假的值

```
var a="abc";  
var b=true;  
var c=false;  
var d=null;
```

第一种情况: `if(a && b)` 返回为`true`

`if(b && a)` 返回为`abc`

第二种情况: `if(c && d)` 返回`false`

`if(d && c)` 返回为`null`

或运算:

1. 当表达式全为假, 返回最后一个表达式的值

2. 只要有一个表达式为真, 返回第一个真的值

第一种: `if(d || c)` 返回`false`

`if(c || d)` 返回`null`

第二种: `if(a || b)` 返回`true`

`if(c || a)` 返回`abc`

数组

格式

```
var 数组名 = []  
var 数组名 = [123, "123", true]
```

在JavaScript中, 只要我们通过数组下标赋值, 取最大的下标扩容 //遍历数组

```
for(var i=0; i<a.length; i++)  
{  
    alert(a[i])  
}
```

函数

注意:

JavaScript中函数不能重载, 如果重名, 下面的函数将覆盖上面的函数

第一种方式

```

function 函数名(形参列表)
{
    函数体;
}
//无参函数
function fun1(){
    alert("我是无参函数");
}
//带参函数并有返回值
function fun2(a,b){
    return a+b;
}

```

第二种方式

```

var 变量名 = function(形参列表){
    函数体;
}
//无参函数
var fun1 = function(){
    alert("我是无参函数");
}
//带参函数并有返回值 var fun2 = function(a,b){
    return a+b;
}

```

函数的arguments隐性参数

```

//虽然是无参的,但是有隐形的参数
function func(){
    alert(arguments[0]); //输出1
    alert(arguments[1]); //输出2
    alert(arguments[2]); //输出3

    for(var i=0;i<arguments.length;i++){
        alert(arguments[i]); //输出1 2 3
    }
}
func(1,2,3);

```

对象的定义

方式一:通过newObject

```
var obj = new Object();
obj.name = "马佳盛";
obj.age = 18;
obj.fun = function(){
    alert("名字是:"+this.name+"年龄是:"+this.age)
}
obj.fun();
```

方式二:直接{}

```
var obj = {
    name:"马佳盛",
    age:18,
    info : function(){ alert("名字是:"+this.name+"年龄是:"+this.age) } };
obj.info();
```

事件

静态注册

静态:狗都不用 通过html标签直接给标签赋值事件,然后在js中直接写时间的方法

动态注册

通过js代码获取标签dom,然后通过 dom.事件名 = function(){}

常用的事件

onload:页面加载完成后执行,常用于页面初始化
onclick:单机事件
onblur:失去焦点事件
onchange:内容发生改变事件
onsubmit:表单提交事件

onload

静态：

```
<body onload="load();">
<script>
    function load(){
        alert("加载完成"); }
</script>
</body>
```

动态：

```
window.onload = function(){
    alert("动态注册onload事件");
}
```

onclick

静态：

```
<button onclick="click1();">按钮</button>
<script>
    function click1(){
        alert("加载完成");
    }
</script>
```

动态：

```
<button id="button1">按钮</button>
<script>
    var but = document.getElementById("button1");
    but.onclick = function(){ alert("我是按钮"); } </script>
```

以下只有动态注册

onblur

```
账户:<input type="text" name="" id="zhanghu">
密码:<input type="text" name="" id="mima">
<script>
    var user = document.getElementById("zhanghu");
    var pass = document.getElementById("mima");
    user.onblur = function(){ alert("账户失去了焦点!!!"); }
    pass.onblur = function(){ alert("密码失去了焦点!!!"); }
</script>
```

onchange

```

<select name="" id="sele">
  <option value="">---名字---</option>
  <option value="">张三</option>
  <option value="">李四</option>
  <option value="">王五</option>
</select>
<script>
  var sele = document.getElementById("sele");
  sele.onchange = function(){ alert("内容发生了改变!!!"); }
</script>

```

onsubmit

```

<form action="http://localhost:8080" id="forms">
  <input type="submit" value="动态注册">
</form>
<script>
var submit = document.getElementById("forms"); submit.onsubmit =
function(){
  alert("没有通过");
  return false; //当onsubmit返回false的时候表示不提交
}
</script>

```

Dom

getElementById():

```

document.getElementById("id名");
//获取标签为id名的标签,只返回第一个

```

案例:验证文本框中字符串是否由是字母、数字、下划线,至少5-12位

```

<input type="text" name="" id="text1" value="dddd"> <span id="spa"
style="color: red;"></span>

<input type="button" value="提交" id="but">

<script>
var text = document.getElementById("text1");
var but = document.getElementById("but");
var spa = document.getElementById("spa");
but.onclick = function () {
  var str = text.value;
  //获取文本框的值 var patt = /^w{5,12}$/; //检查是否5-12位

  if (patt.test(str)) { //字符串是否匹配规则
    spa.innerHTML = "合法";
  }
}

```

```
    } else{
        spa.innerHTML = "不合法"; }
    }
</script>
```

getElementsName

```
document.getElementsByName("name名");
//返回所有name名的标签
```

案例:实现按钮全选、全不选、反选

```
<input type="checkbox" name="check" id="" value="java">java
<input type="checkbox" name="check" id="" value="php">php
<input type="checkbox" name="check" id="" value="python">python
<button id="all">全选</button>
<button id="noall">全不选</button>
<button id="revers">反选</button>
<script>
    var list=document.getElementsByName("check");
    var but1 = document.getElementById("all");
    var but2 = document.getElementById("noall");
    var but3 = document.getElementById("revers");
    but1.onclick = function(){
        for(vari=0;i<checklist.length;i++){
            checklist[i].checked = true;
        }
    }

    but2.onclick = function(){
        for(vari=0;i<checklist.length;i++)
        {
            checklist[i].checked = false;
        }
    }

    but3.onclick = function(){
        for(vari=0;i<checklist.length;i++){
            if(checklist[i].checked==true){
                checklist[i].checked=false;
            }else{
                checklist[i].checked = true;
            }
        }
    }
}
```

getElementsByTagName

获取所有input标签

```
document.getElementsByTagName("input");
```

方法

```
//查询city下的所有li节点  
getElementById("city").getElementsByTagName("li");  
//查询city下的所有子节点 getElementById("city").childNodes.length;  
//查询city下的第一个子节点 getElementById("city").firstChild;  
//返回city字节的父节点 document.getElementById("city").parentNode;  
//返回city前面的兄弟节点  
document.getElementById("city").previousSibling;
```

正则表达式

```
声明一个表达式 var str = /[a-z]/;  
[abc] 字符串是否包含abc  
[a-z] 字符串是否包含小写a-z  
[A-Z] 字符串是否包含大写A-Z  
[0-9] 字符串是否包含0-9  
\w 字符串是否包含 a-z A-Z 0-9 下划线  
\W 不包含\w的内容  
\d 数字  
\D 非数字  
\s 空白字符  
\S 非空白字符  
\n 查找换行符  
\r 查找回车符  
+ 至少一个  
* 零个或多个  
? 零个或一个  
a{3} 连续3个a  
a{3-5} 连续3-5个a  
a{3,} 大于3的无限个  
a$ 以a结尾的字符串  
^a 以a开始的字符串  
?=a 匹配a前面的字符串
```

jQuery

\$使用

1. \$(函数):表示页面加载完成之后.相当于window.onload = function(){}

```
$(function(){  
  
    alert("加载完成");  
  
})
```

2. \$(HTML字符串):会创建标签

案例:在body中创建2个p标签

```
<script src="jquery-1.7.2.js"></script>  
  
<script>  
  
    $(function(){  
  
        $("<p>今天天气好晴朗!</p>" +  
  
            "<p>处处好风光!</p>").appendTo("body");  
  
    });  
  
</script>
```

3. \$(选择器字符串):

\$("#id"):根据id查询标签

\$("标签名"):根据标签名查询标签

\$(".class"):根据class查询标签

var but1 = \$("#but");

4. \$(dom对象):转换为jQuery对象

Var a = \$(document.getElementById(""));

jQuery转换为dom对象: a[0];

简单使用:单机按钮输出hello world

单机

```
<script src="jquery-1.7.2.js"></script>

<script>

    $(function(){

        var but1 = $("#but");

        but1.click(function(){

            alert("hello world!")

        })

    });

</script>
```

选择器

基本选择器

`$("#div")` //查找id为div的标签

`$("div")` //查找所有div标签

`$(".div")` //查找class为div的标签

`$("*")` //查找所有元素

`$("#div,.div")` //查找id为div, class为div的标签

层级选择器

`$("form input")` //from下的所有input元素`

`$("form > input")` //form下的input子元素`

`$("label + input")` // 所有label的兄弟元素`

`$("form ~ input")` //之后的兄弟元素`

过滤选择器

`$('li:first');` //匹配到的第一个元素

`:last` 获取最后个元素

`:not(selector)` 返回值去掉selector

`:even` 匹配所有索引值为偶数的元素, 从 0 开始计数

`:odd` 匹配所有索引值为奇数的元素, 从 0 开始计数

`:eq(index)` 返回一个数组,准确获取那个元素

`:gt(index)` 匹配所有大于给定索引值的元素

`:lt(index)` 匹配所有小于给定索引值的元素

`:header` 匹配如 h1,h2,h3 之类的标题元素

`:animated` 匹配所有正在执行动画效果的元素

`$("#login:focus");` 当页面加载后将 id 为 'login' 的元素设置焦点:

内容筛选器

`$("#div:contains('John'))` // `` 标签中包含字符串,字符串包含john的标签返回

`$("#td:empty")` // `` 查找没有子元素和标签中没有字符串的标签

`$("#td:parent")` //查找所有子元素和标签中没有字符串的标签

`$("#div:has(p)).addClass("test");` //查询所有p标签,然后给p标签添加class为test

隐藏选择器

`$("#tr:hidden")` // `` 找到隐藏的标签

`$("#tr:visible")` // `` 所有不隐藏的标签

属性过滤器

`$("#div[id]")` // `` 查找有id的div

`$("#input[name='newsletter']")` // 查找所有 name 属性是 newsletter 的 input 元素

`$("input[name!='newsletter']")` // 查找所有 name 属性不是 newsletter 的 input 元素

`$("input[name^='news']")` // 查找所有 name 以 'news' 开始的 input 元素

`$("input[name$='letter']")` // 查找所有 name 以 'letter' 结尾的 input 元素

`$("input[name*='man']")` // 查找所有 name 包含 'man' 的 input 元素

`$("input[id][name$='man']")` // 找到所有含有 id 属性, 并且它的 name 属性是以 man 结尾的

表单过滤器

`$(":input")` 匹配所有 input, textarea, select 和 button 元素

`:text` 匹配所有 文本输入框

`:password` 匹配所有的密码输入框

`:radio` 匹配所有的单选框

`:checkbox` 匹配所有的复选框

`:submit` 匹配所有提交按钮

`:image` 匹配所有 img 标签

`:reset` 匹配所有重置按钮

`:button` 匹配所有 inputtype=button

按钮

`:file` 匹配所有 inputtype=file 文件上传

`:hidden` 匹配所有不可见元素 display:none 或 inputtype=hidden

表单属性过滤器

`$("input:enabled")` 所有可见的元素

`:disabled` 匹配所有不可用元素

`:checked` 匹配所有选中的单选, 复选, 和下拉列表中选中的 option 标签对象

`:selected` 匹配所有选中的 option

jQuery的属性

传参设置, 不穿获取

html()

获取和设置标签中的内容

案例一:修改p标签的字符串为处处好风光

今天天气好晴朗

```
<script src="jquery-1.7.2.js"></script>

<script>

$(function(){

    $("#p1").html("<h1>处处好风光</h1>");

});

</script>
```

text()

获取和设置标签中的文本

案例一:修改p标签的文本为哈哈

今天天气好晴朗

```
<script src="jquery-1.7.2.js"></script>

<script>

$(function(){

    $("#p1").text("哈哈");

});

</script>
```

prop()

与attr取反,可用于任何返回true false的属性

alert(\$("#check1").prop("checked")); //获取多选框的选中状态

val()

获取和设置标签单的value

第一个作用:设置value的值

案例一:修改button标签的value为点击一下

```
<input type="button" value="aaaa" id="but1">

<script src="jquery-1.7.2.js"></script>

    <script>

        $(function(){

            $("#but1").val("点击一下");

        });

    </script>
```

第二个作用:设置单选、多选、下拉多选,下拉单选的选中状态只需要在val(["1","2"])

```
<input type="radio" name="gender" id="rad1" value="ra1">

<input type="radio" name="gender" id="rad2" value="ra2">

<input type="checkbox" name="che1" id="" value="che1"
type="checkbox1">

<input type="checkbox" name="che2" id="" value="che2"
type="checkbox1">

<script src="jquery-1.7.2.js"></script>

    <script>

        $(function(){

            $(":radio").val(["ra1"]); //选中单选

            $(":checkbox").val(["che1","che2"]); //选中多选

        });

    </script>
```

一下操作多个标签

```
$(":radio,:checkbox,.....").val(["che1","che2",value值]);
```

attr()

获取设置标签的属性,但不推荐获取true、false的属性,比如说单选多选框的选中状态

第一种使用:获取和设置属性值

```
alert($("#rad1").attr("name")); //获取name
```

```
alert($("#rad1").attr("name","男或女")); //设置name
```

第二种使用:给标签一个自定义属性

```
alert($("#rad1").attr("aaa","自定义值")); //设置一个aaa的属性
```

```
alert($("#rad1").attr("aaa")); //获取一个aaa的属性
```

jQuery实现全选、全不选、反选

```
<form method="post" action="">
```

你爱好的运动是? <input type="checkbox" id="checkedAllBox" />全选/全不选

```
<br />
```

```
<input type="checkbox" name="items" value="足球" />足球
```

```
<input type="checkbox" name="items" value="篮球" />篮球
```

```
<input type="checkbox" name="items" value="羽毛球" />羽毛球
```

```
<input type="checkbox" name="items" value="乒乓球" />乒乓球
```

```
<br />
```

```
<input type="button" id="checkedAllBtn" value="全 选" />
```

```
<input type="button" id="checkedNoBtn" value="全不选" />
```

```
<input type="button" id="checkedRevBtn" value="反 选" />
```

```
<input type="button" id="sendBtn" value="提 交" />
```

```
</form>
```

```
<script type="text/javascript" src="jquery-1.7.2.js"></script>
```

```
<script type="text/javascript">
```

```
$(function(){
```

```
    $("#checkedAllBtn").click(function(){
```

```
        $(":checkbox[name=items]").prop("checked",true);
```

```
    });
```

```
    $("#checkedNoBtn").click(function(){
```

```
        $(":checkbox[name=items]").prop("checked",false);
```

```
    });
```

```
    $("#checkedRevBtn").click(function(){
```

```
        item = $(":checkbox[name=items]").each(function(){
```

```
            this.checked = !this.checked
```

```
        });
```

```
        var count1 = $(":checkbox[name=items]").length;
```

```
        var count2 = $(":checkbox[name=items]:checked").length;
```

```

        console.log(count2);
        if(count1 == count2){
            $("#checkedAllBox").prop("checked", true)
        }else{
            $("#checkedAllBox").prop("checked", false)
        }
    });
    $("#checkedAllBox").click(function(){
        if($("#checkedAllBox").prop("checked")){
            $(".checkbox[name=items]").prop("checked", true);
        }else{
            $(".checkbox[name=items]").prop("checked", false);
        }
    });
});
</script>

```

标签的增、删、改

增加

`$("#p").append("Hello");` ``把b标签增加到p标签中``

`Dom1.appendTo(dom2)` 把dom1插入到dom2所有子元素后面

`Dom1.prependTo(dom2)` 把dom1插入dom2所有子元素的前面

`Dom1.insertAfter(dom2)` 把dom2插入到dom1的前面

`Dom1.insertBefore(dom2)` 把dom2插入到dom1的后面

替换

`Dom1.replaceWith(dom2)` 把dom1替换为dom2

`Dom1.replaceAll(dom2)` 把所有dom1替换为dom2

删除

`Dom1.remove()` 删除dom1, 标签也删除

`Dom1.empty()` 清空dom1的内容, 标签还在

案例一:多选框的内容从左到右,从右到左

Body

```

<div id="left">

    <select multiple="multiple" name="se101">

```



```

        <option value="opt01">选项1</option>
        <option value="opt02">选项2</option>
        <option value="opt03">选项3</option>
        <option value="opt04">选项4</option>
        <option value="opt05">选项5</option>
        <option value="opt06">选项6</option>
        <option value="opt07">选项7</option>
        <option value="opt08">选项8</option>
    </select>

    <button>选中添加到右边</button>

    <button>全部添加到右边</button>

</div>

<div id="righth">

    <select multiple="multiple" name="sel02">

    </select>

    <button>选中删除到左边</button>

    <button>全部删除到左边</button>

</div>

```

Css

```

<style type="text/css">

    •   select {

    •       width: 100px;

    •       height: 140px;

    •   }

    •

    •   div {

    •       width: 130px;

    •       float: left;

```

- text-align: center;
- }

</style>

Script

```
<script type="text/javascript" src="jquery-1.7.2.js"></script>

<script type="text/javascript">

    $(function(){

        $("button:eq(0)").click(function(){

            $("select:eq(0)
option:selected").appendTo($("select:eq(1)"));

        });

        $("button:eq(1)").click(function(){

            $("select:eq(0) option").appendTo($("select:eq(1)"));

        });

        $("button:eq(2)").click(function(){

            $("select:eq(1)
option:selected").appendTo($("select:eq(0)"));

        });

        $("button:eq(3)").click(function(){

            $("select:eq(1) option").appendTo($("select:eq(0)"));

        });

    });

</script>
```

案例二:动态添加删除表格记录

Body

```
<table id="employeeTable">
  <tr>
    •   <th>Name</th>

    •   <th>Email</th>

    •   <th>Salary</th>

    •   <th>&nbsp;</th>

    •   </tr>

    •   <tr>

    •   <td>Tom</td>

    •   <td>tom@tom.com</td>

    •   <td>5000</td>

    •   <td><a href="deleteEmp?id=001">Delete</a></td>

    •   </tr>

    •   <tr>

    •   <td>Jerry</td>

    •   <td>jerry@sohu.com</td>

    •   <td>8000</td>

    •   <td><a href="deleteEmp?id=002">Delete</a></td>

    •   </tr>

    •   <tr>

    •   <td>Bob</td>

    •   <td>bob@tom.com</td>

    •   <td>10000</td>

    •   <td><a href="deleteEmp?id=003">Delete</a></td>
```

- `</tr>`

`</table>`

`<div id="formDiv">`

- `<h4>添加新员工</h4>`

`<table>`

- `<tr>`

- `<td class="word">name: </td>`

- `<td class="inp">`

- `<input type="text" name="empName" id="empName" />`

- `</td>`

- `</tr>`

- `<tr>`

- `<td class="word">email: </td>`

- `<td class="inp">`

- `<input type="text" name="email" id="email" />`

- `</td>`

- `</tr>`

- `<tr>`

- `<td class="word">salary: </td>`

- `<td class="inp">`

- `<input type="text" name="salary" id="salary" />`

- `</td>`

- `</tr>`

- `<tr>`

- `<td colspan="2" align="center">`

- `<button id="addEmpButton" value="abc">`
- `Submit`
- `</button>`
- `</td>`
- `</tr>`
- `</table>`
- `</div>`

Css

```
<style>

• \#total {

• width: 450px;

• margin-left: auto;

• margin-right: auto;

• }

• ul {

• list-style-type: none;

• }

• li {

• border-style: solid;

• border-width: 1px;

• padding: 5px;

• margin: 5px;
```

```
• background-color: #99ff99;

• float: left;

• }

• .inner {

• width: 400px;

• border-style: solid;

• border-width: 1px;

• margin: 10px;

• padding: 10px;

• float: left;

• }

• \#employeeTable {

• border-spacing: 1px;

• background-color: black;

• margin: 80px auto 10px auto;

• }

• th,

• td {

• background-color: white;

• }

• \#formDiv {
```

```

• \#formDiv {
    width: 250px;

    border-style: solid;

    border-width: 1px;

    margin: 50px auto 10px auto;

    padding: 10px;

}

• \#formDiv input {

    width: 100%;

}

• .word {

    width: 40px;

}

• .inp {

    width: 200px;

}

</style>

```

Script

```

<script type="text/javascript" src="jquery-1.7.2.js"></script>

<script type="text/javascript">

• $(function () {

```

```
•     var remo = function (event) {  
  
•         //获取tr  
  
•         var hang1 = $(this).parent().parent();  
  
•         var txt = hang1.find("td:first").text();  
  
•         if (confirm("确定要删除"+txt+"嘛?")) {  
  
•             hang1.remove();  
  
•         }  
  
•         return false;  
  
•     };  
  
•     //添加  
  
•     $("#addEmpButton").click(function () {  
  
•         var name = $("#empName").val();  
  
•         var email = $("#email").val();  
  
•         var salary = $("#salary").val();  
  
•         var $tianjia = $("<tr>" +  
  
•             "<td>" + name + "</td>" +  
  
•             "<td>" + email + "</td>" +  
  
•             "<td>" + salary + "</td>" +  
  
•             "<td><a href=\"deleteEmp?id=001\">Delete</a></td>" +  
  
•             "</tr>");  
  
•         $tianjia.appendTo("#employeeTable");  
  
•         $tianjia.find("a").click(remo);  
  
•     });  
  
•     //删除
```


- `$("#a").click(remo);`
- `});`

`</script>`

CSS样式操作

AddClass() 添加样式

Remove() 删除样式

toggleClass() 有就删除,没有添加

offset() 获取和设置元素的坐标

案例:点击按钮对div进行设置

HTML

```
<table align="center">

•   <tr>

•   <td>

        <div class="border">

•   </div>

•   </td>

•

•   <td>

        <div class="btn">

•   <input type="button" value="addClass()" id="btn01"/>

•   <input type="button" value="removeClass()" id="btn02"/>

•   <input type="button" value="toggleClass()" id="btn03"/>

•   <input type="button" value="offset()" id="btn04"/>

•   </div>

•   </td>
```

- </tr>

</table>

CSS

div{

- width:100px;

- height:260px;

}

div.border{

- border: 2px white solid;

}

div.redDiv{

- background-color: red;

}

div.blackDiv{

- border: 5px blue solid;

}

Script

<script type="text/javascript" src="jquery-1.7.2.js"></script>

<script type="text/javascript">

```
<script type= text/javascript >
```

- `$(function(){`
- `var $div = $("div:eq(0)");`
- `//添加红色背景蓝色边框`
- `$("#btn01").click(function(){`
- `$div.addClass('redDiv blackDiv');`
- `});`
- `//删除红色背景,如果不传入参数,则全部删除`
- `$("#btn02").click(function(){`
- `$div.removeClass('redDiv');`
- `});`
- `//删除红色背景,如果不传入参数,则全部删除`
- `$("#btn03").click(function(){`
- `$div.toggleClass('redDiv');`
- `});`
- `//获取和设置div的坐标,距离页面位置`
- `$("#btn04").click(function(){`
- `var size =$div.offset(); //获取div的位置`
- `$div.offset({`
- `top:100,`
- `left:50`

```
• });  
  
• });  
  
• });  
  
</script>
```

动画:

案例一:给div添加动画效果

HTML

```
<table style="float: left;">  
  
• <tr>  
  
• <td><button id="btn1">显示show()</button></td>  
  
• </tr>  
  
• <tr>  
  
• <td><button id="btn2">隐藏hide()</button></td>  
  
• </tr>  
  
• <tr>  
  
• <td><button id="btn3">显示/隐藏切换 toggle()</button></td>  
  
• </tr>  
  
• <tr>  
  
• <td><button id="btn4">淡入fadeIn()</button></td>  
  
• </tr>  
  
• <tr>  
  
• <td><button id="btn5">淡出fadeOut()</button></td>  
  
• </tr>  
  
• <tr>
```

- `<td><button id="btn6">淡化到fadeTo()</button></td>`
- `</tr>`
- `<tr>`
- `<td><button id="btn7">淡化切换fadeToggle()</button></td>`
- `</tr>`
- `</table>`
- `<div id="div1" style="float:left;border: 1px solid;background-color: blue;width: 300px;height: 200px;">`
- jquery动画定义了很多种动画效果，可以很方便的使用这些动画效果
- `</div>`

Script

```
<script type="text/javascript" src="jquery-1.7.2.js"></script>
```

```
<script type="text/javascript">
```

- ``//第一个参数为响应时间,第二个参数可以加函数``
- ``$(function () {``
- ``//显示 show()``
- ``$("#btn1").click(function () {``
- ``$("#div1").show(1000);``
- ``});``
- ``//隐藏 hide()``
- ``$("#btn2").click(function () {``
- ``$("#div1").hide(1000);``
- ``});``
- ``//切换 toggle()``

```

// 单击 toggle()

•   `$("#btn3").click(function () {`

•   `$("#div1").toggle(1000);`

•   `});`

` `

•   `//淡入 fadeIn()`

•   `$("#btn4").click(function () {`

•   `$("#div1").fadeIn(500);`

•   `});`

•   `//淡出 fadeOut()`

•   `$("#btn5").click(function () {`

•   `$("#div1").fadeOut(500);`

•   `});`

` `

•   `//参数:第一个为时间,第二个为透明度`

•   `//淡化到 fadeTo()`

•   `$("#btn6").click(function () {`

•   `$("#div1").fadeTo(1000, Math.random());`

•   `});`

` `

•   `//淡化切换 fadeToggle()`

•   `$("#btn7").click(function () {`

•   `$("#div1").fadeToggle(1000, "linear");`

•   `});`

•   `}``

` `

```

</script>

案例二:产品展示

HTML

```
<div class="SubCategoryBox">

  <ul>

    <li><a href="#">佳能</a><i>(30440) </i></li>

    <li><a href="#">索尼</a><i>(27220) </i></li>

    <li><a href="#">三星</a><i>(20808) </i></li>

    <li><a href="#">尼康</a><i>(17821) </i></li>

    <li><a href="#">松下</a><i>(12289) </i></li>

    <li><a href="#">卡西欧</a><i>(8242) </i></li>

    <li><a href="#">富士</a><i>(14894) </i></li>

    <li><a href="#">柯达</a><i>(9520) </i></li>

    <li><a href="#">宾得</a><i>(2195) </i></li>

    <li><a href="#">理光</a><i>(4114) </i></li>

    <li><a href="#">奥林巴斯</a><i>(12205) </i></li>

    <li><a href="#">明基</a><i>(1466) </i></li>

    <li><a href="#">爱国者</a><i>(3091) </i></li>

    <li><a href="#">其它品牌相机</a><i>(7275) </i></li>

  </ul>

  <div class="showmore">

    <a href="more.html"><span>显示全部品牌</span></a>

  </div>

</div>
```

CSS

```
<style type="text/css">

* {

    margin: 0;

    padding: 0;

}

body {

    font-size: 12px;

    text-align: center;

}

a {

    color: #04D;

    text-decoration: none;

}

a:hover {

    color: #F50;

    text-decoration: underline;

}

.SubCategoryBox {

    width: 600px;
```



```
margin: 0 auto;

text-align: center;

margin-top: 40px;
}

.SubCategoryBox ul {

list-style: none;
}

.SubCategoryBox ul li {

display: block;

float: left;

width: 200px;

line-height: 20px;
}

.showmore , .showless{

clear: both;

text-align: center;

padding-top: 10px;
}

.showmore a , .showless a{

display: block;

width: 120px;
```

```
margin: 0 auto;

line-height: 24px;

border: 1px solid #AAA;

}

.showmore a span {

padding-left: 15px;

background: url(img/down.gif) no-repeat 0 0;

}

.showless a span {

padding-left: 15px;

background: url(img/up.gif) no-repeat 0 0;

}

.promoted a {

color: #F50;

}

</style>
```

Script

```
<script type="text/javascript" src="jquery-1.7.2.js"></script>

<script type="text/javascript">

$(function() {

•   var li5 = $(("ul:first li:gt(5):not(:last)"));
```

```

    • li5.hide();

    • $("div div a").click(function(){

    •

    • if(li5.is(":hidden")){

    •     li5.show();

    •     $("div div").removeClass();

    •     $("div div a span").text("精简产品!")

    •     $("div div").addClass("showless");

    •     $("li:contains('索尼')").addClass("promoted")

    •     $("li:contains('卡西欧')").addClass("promoted")

    •     $("li:contains('富士')").addClass("promoted")

    • }else{

    •     li5.hide();

    •     $("div div a span").text("全部产品!")

    •     $("div div").removeClass();

    •     $("div div").addClass("showmore");

    •     $("li").removeClass();

    • }

    • return false;

    • });

    • });

</script>

```

jQuery的事件

案例一:对H5标签设置事件

事件冒泡:如果给子元素加了click事件,父元素的click也会执行,如果不想让父元素执行,在函

数中增加 return false;

HTML

```
<div id="panel">
```

- ```
<input type="text" name="" id="text">
```

- ```
<h5 class="head">什么是jQuery?</h5>
```

```
<div class="content">
```

- jQuery是继Prototype之后又一个优秀的JavaScript库，它是一个由 John Resig 创建于2006年1月的开源项目。jQuery凭借简洁的语法和跨平台的兼容性，极大地简化了JavaScript开发人员遍历HTML文档、操作DOM、处理事件、执行动画和开发Ajax。它独特而又优雅的代码风格改变了JavaScript程序员的设计思路和编写程序的方式。

- ```
</div>
```

- ```
<button>按钮</button>
```

- ```
</div>
```

## CSS

```
#####
```

```
<style>
```

- ```
\* {
```

- ```
margin: 0;
```

- ```
padding: 0;
```

- ```
}
```

- ```
body {
```

- ```
font-size: 13px;
```

- ```
line-height: 130%;
```

- ```
padding: 60px;
```

- ```
}
```

- \#panel {
- width: 300px;
- border: 1px solid #0050D0;
- }

- .head {
- padding: 5px;
- background: #96E555;
- cursor: pointer;
- }

- .content {
- padding: 10px;
- text-indent: 2em;
- border-top: 1px solid #0050D0;
- display: block;
- display: none;
- }

- .highlight {
- background: #FF3300
- }

- </style>

Script

blur() 失去焦点

```
$(function () {
```

- \$("#text").blur(function(){
- alert("失去焦点了!");
- });
- });

click() 单机事件

```
$("#h5").click(function(){
```

- alert("点击了h5");
- });

mouseover() 鼠标移入

```
$("#h5").mouseover(function(){
```

- alert("鼠标放入了h5");
- });

mouseout() 鼠标移除

```
$("#h5").mouseout(function(){
```

- alert("鼠标移出了h5");
- });

bind() 可以给标签添加多个事件

```
$("#h5").bind("click mouseout",function(){
```

- alert("开启多个事件");
- });

可以一次给一个事件只能使用一次

One() 跟bind一样,但是每一个事件只能使用一次

```
$("#h5").one("click mouseout",function(){
```

- alert("开启多个事件"); //单机一次 移出一次
- });

Unbind() 删除bind绑定的事件

Live() 和bind使用方法一样,什么时候设置的事件都有效

比如说:有一个h5添加了一个事件,在他之后又添加了一个h5,第二个h5就没有事件

如果用了live就自动的给h5添加了事件

获取事件的相关信息

在事件函数的参数中增加一个参数

```
$("#h5").click(function(event){
```

- console.log(event); //单机一次 移除一次
- });

案例二:放大镜

HTML

```

```

```
    <div id="showBig">
```

```
        
```

```
    </div>
```

CSS

#####

```
<style type="text/css">
```

```
body {  
  
    text-align: center;  
  
}  
  
\#small {  
  
    margin-top: 150px;  
  
}  
  
\#showBig {  
  
    position: absolute;  
  
    display: none;  
  
}
```

Script

```
#####

<script type="text/javascript" src="jquery-1.7.2.js"></script>

<script type="text/javascript">

    $(function(){

•     var dashow = $("#showBig");

•     $("#small").bind("mouseover mouseout mouseover",function(ele){

•         if(ele.type == "mouseover"){

•             dashow.show();

•         }else if(ele.type == "mouseout"){

•             dashow.hide();

•         }else if(ele.type == "mouseover"){

•             console.log(ele);


```



```
•      dashow.offset({  
  
•      top:ele.pageX,  
  
•      left:ele.pageY  
  
•      });  
  
•      }  
  
•      });  
  
      });  
</script>
```