# Ajax简介

Asynchronous javascript and xml

允许浏览器与服务器通信而无须刷新当前页面的技术

传统技术是浏览器把整个页面提交到服务器，Ajax是把数据提交到服务器。

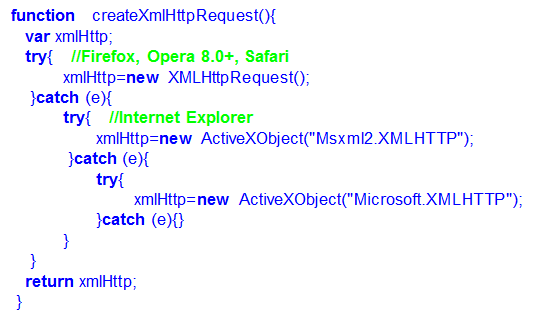
# 实现Ajax

## XmlHttpRequest

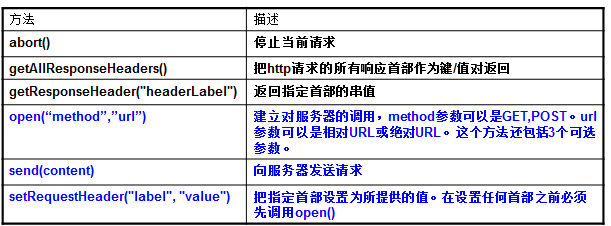
### 介绍



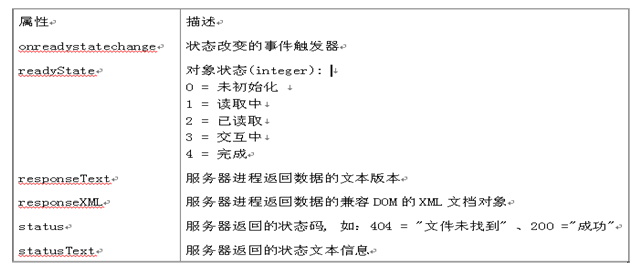
### 初始化



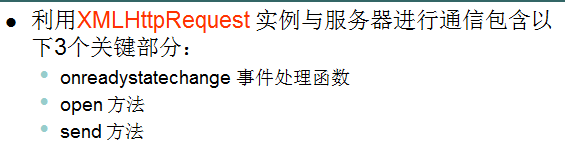
### 对象方法

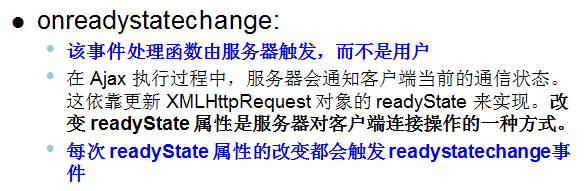


### 对象属性

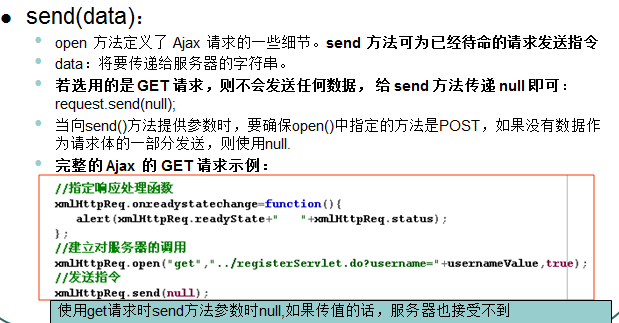


### 发送请求

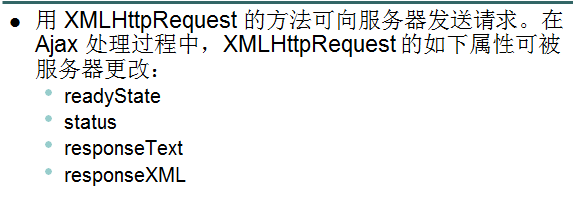


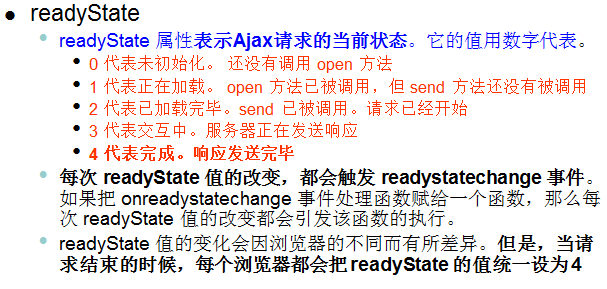


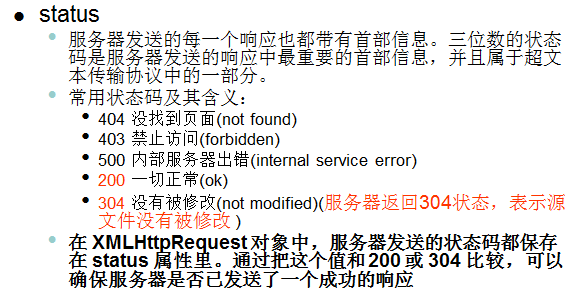




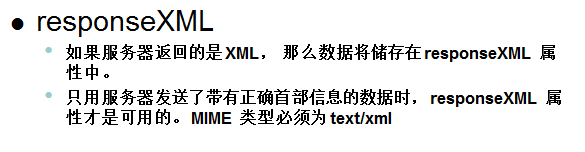




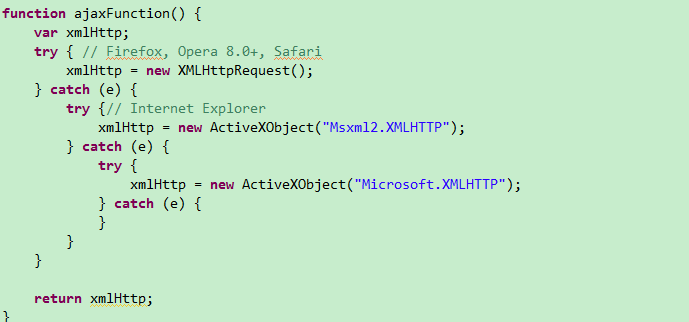










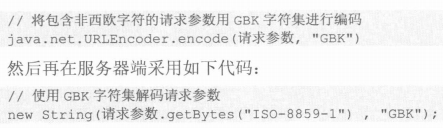


### get和post

* 最好使用post方式
* get一般用于从服务器上获取数据，post请求用于向服务器发送数据。
* get请求传送的数据量较小，一般不能大于2K，post传送的数据量较大
* get的请求参数需要加载url地址之后，post请求通过http post机制，将请求参数以及对应的值放在html header中传输，用户看不到明码的请求参数，post也可以把参数写在url后。
* 中文乱码问题，异步post请求默认采用utf-8字符集编码，设置

request.setCharacterEncoding(“utf-8”)即可获得正确的请求参数

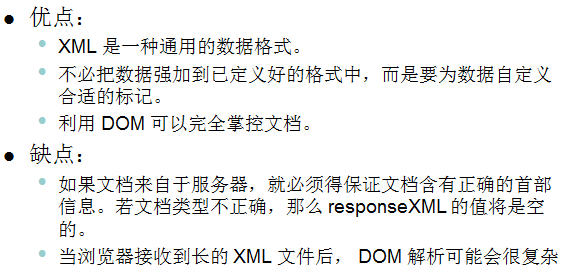
get请求需要在浏览器编码，服务器端解码



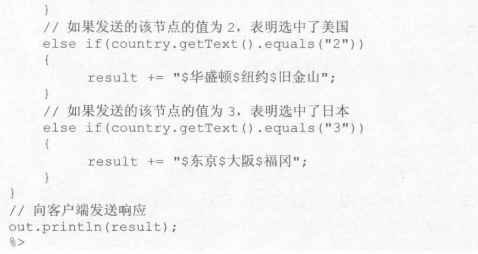
## xml和text优缺点

### xml格式优缺点

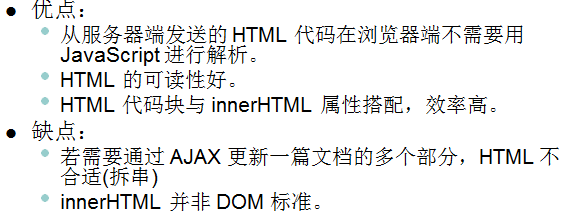
可以用dom解析，服务器端用dom4j处理xml文件





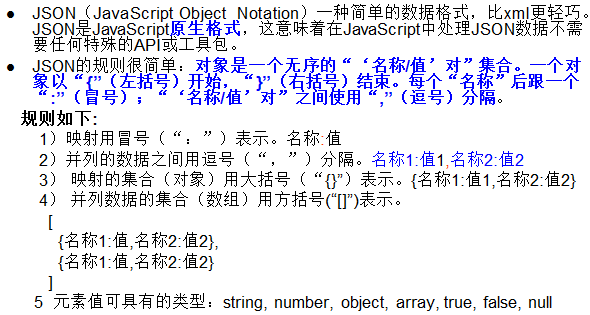


### text格式优缺点

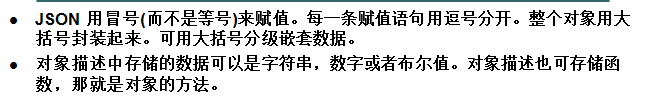


# Json

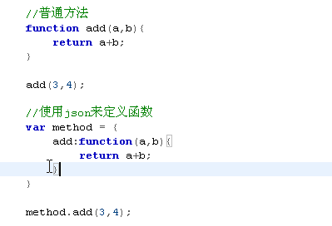
## 简介



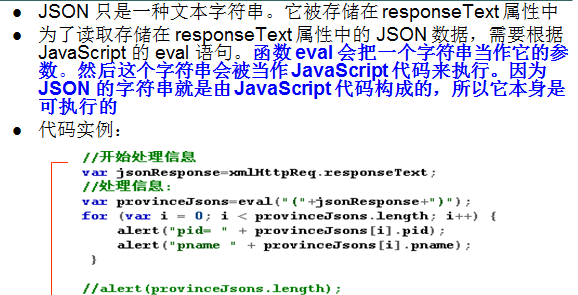


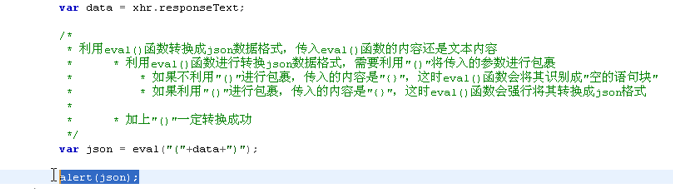


定义方法



## 解析json数据





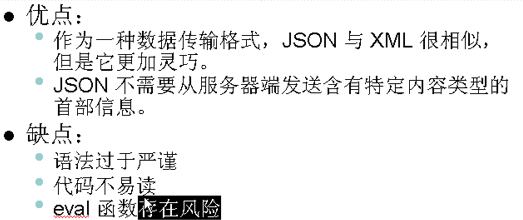
## 服务器端构建json数据格式

引入jar包

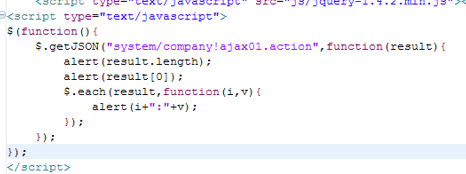


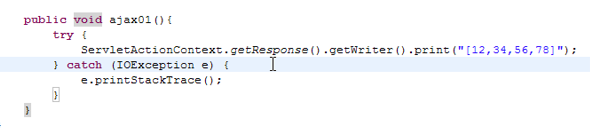


## json优缺点



## 例子





## jsontools把其他数据类型转化成json格式

引入jar包jsontools-core-1.7.jar

**public** **void** ajax06(){

List vo = **new** ArrayList();

**for**(**int** i=0; i<10; i++){

Company c = **new** Company();

c.setName("公司"+i);

Set children = **new** HashSet();

**for**(**int** j=0; j<3; j++){

Department d = **new** Department();

d.setName(c.getName()+"下的部门"+j);

children.add(d);

}

c.setChildren(children);

vo.add(c);

}

**try** {

//[{name:xxx,address:xxx,children:[{name:xxx...},{...}]},{},{},....]

//JSONMapper.toJSON()把其他格式数据转化成json格式，render表示输出格式是否美观易读

String jsonString = JSONMapper.*toJSON*(vo).render(**true**);

System.*out*.println(jsonString);

ServletActionContext.*getResponse*().setCharacterEncoding("UTF-8");

ServletActionContext.*getResponse*().getWriter().print(jsonString);

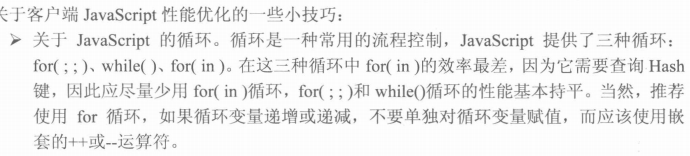
} **catch** (Exception e) {

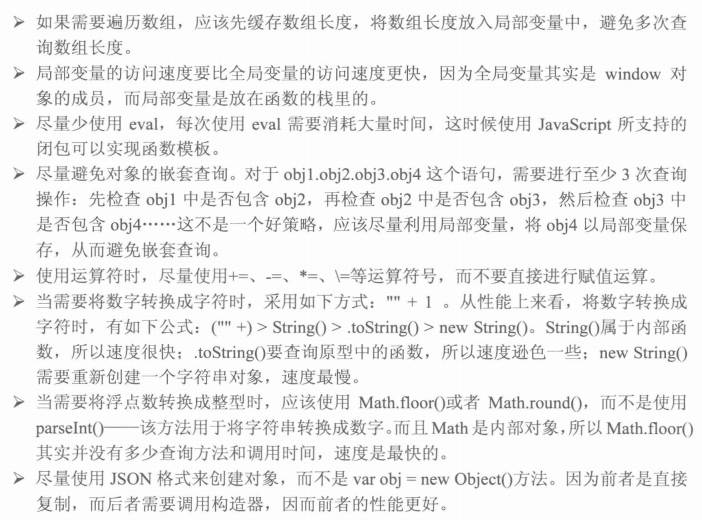
e.printStackTrace();

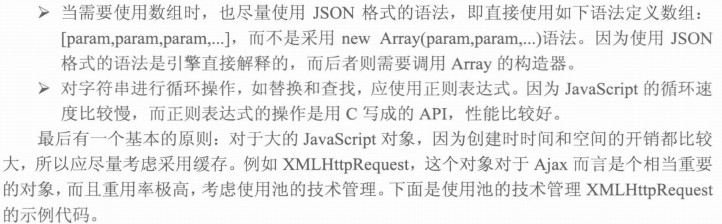
}

}

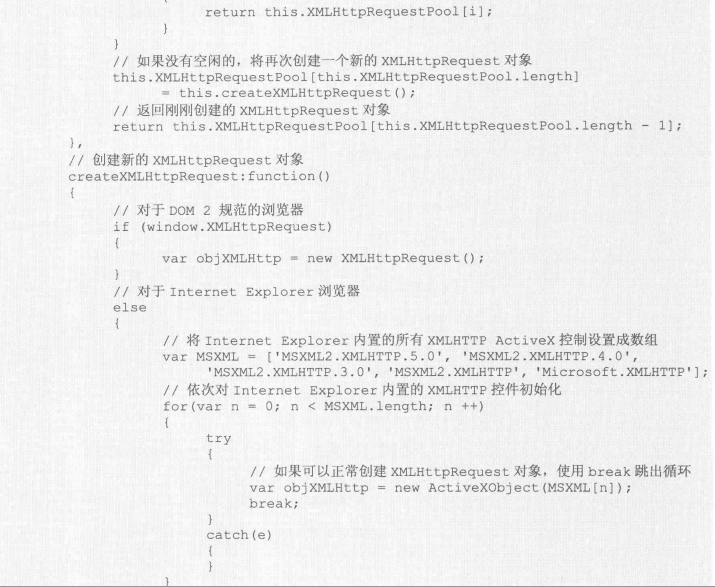
# 性能提升



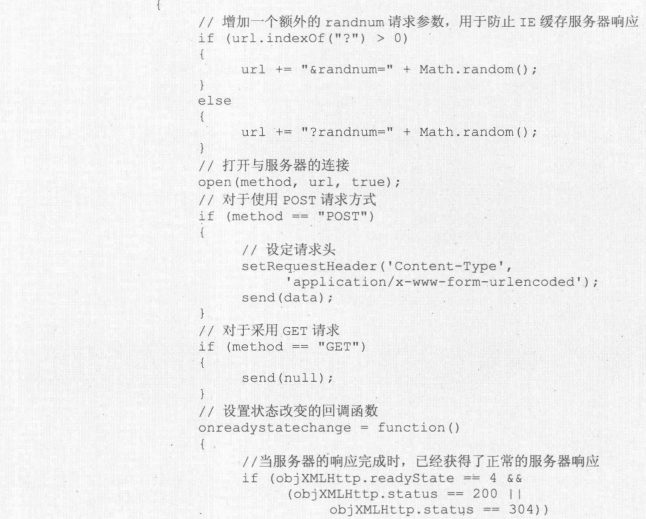


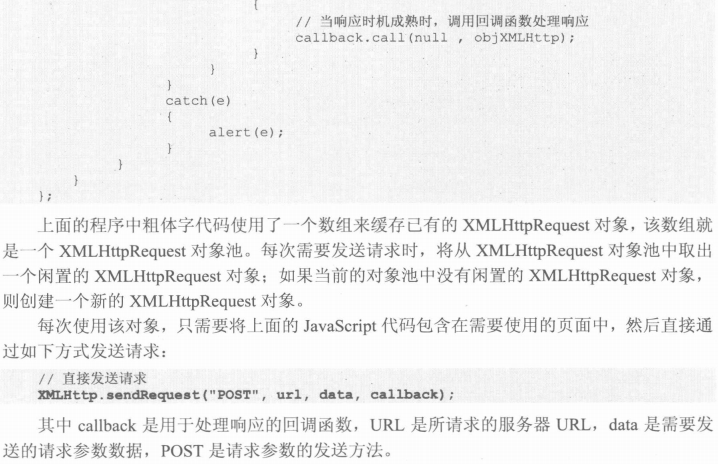












# Ajax内容总结

