

AJAX与JSON数据格式

本章内容：共2小节

- 第1节：AJAX
- 第2节：jquery AJAX API应用

本章目标

- 掌握ajax的基本语法
- 熟练掌握jQuery AJAX API框架的使用方法

第8节 【AJAX】

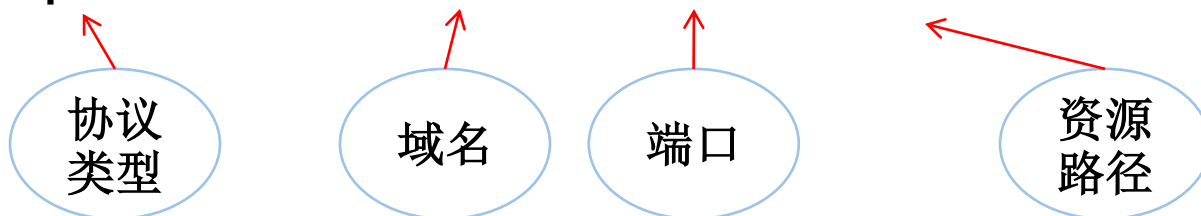
- 知识点1：浏览器与服务器的交互方式
- 知识点2：http请求方式
- 知识点3：服务器响应方式
- 知识点4：AJAX基本概念
- 知识点5：XHR对象
- 知识点6：发送ajax请求
- 知识点7：接收服务器响应
- 知识点8：传输格式
- 知识点9：JSON文件

知识点1- 浏览器与服务器的交互方式-1

- 客户端与服务器交互的方式：

- 浏览器通过用户输入的网址发送http请求。如下为百度首页完整地址：

http://www.baidu.com:80/index.html



- 服务器软件接收http请求，解析客户端传输内容，执行服务器程序进行响应。如下是常用的web服务器以及对应的编程语言。

IIS

Windows自
带免费服务
器，适合asp
项目开发

Apache

应用最广，
免费开源，
支持多语言
(php、java
等)

tomcat

Apache子项
目，适合jsp
项目开发，
一般在开发
以及测试时
使用

weblogic

BEA的产品，
java应用服务
器，提供稳
定安全的服
务器功能，
收费

nodejs

提供了一组接口
来处理网络请求，
第三方组件如
express也实现处
理请求的功能

知识点1- 浏览器与服务器的交互方式-2

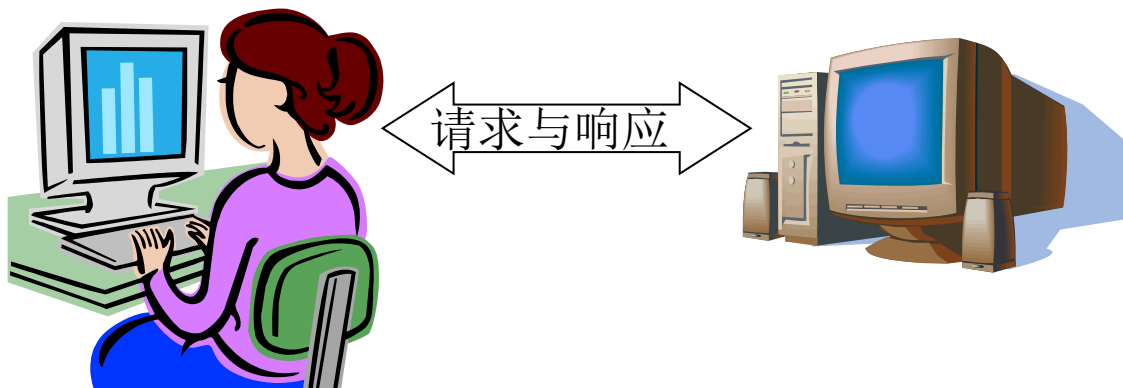
- 用户通过浏览器查阅响应结果，常见的响应结果如下：

网页资源查
阅

网站信息
维护

网络资
源下载

视频音
频播放



知识点2- http请求方式-1

- 浏览器支持http请求方式与服务器交互，浏览器触发http请求的方式

- 使用表单提交

- html的form表单提交

```
<form action="url" method="get">  
  <input type="text" name="userName" />  
  <input type="submit" value="提交"/>  
</form>
```

url: http请求地址

method: 请求类型

请求参
数名

- 通过js提交form表单

```
document.getElementById("myForm").submit();
```

知识点2- http请求方式-2

- 通过改变浏览器的访问地址发送http请求(都是以get请求方式发送)

- 输入地址栏
- 超链接：通过超链接打开新网页

`切换地址`

url: http请求地址

key: 传递参数的参数名

value: 参数值, 字符类型

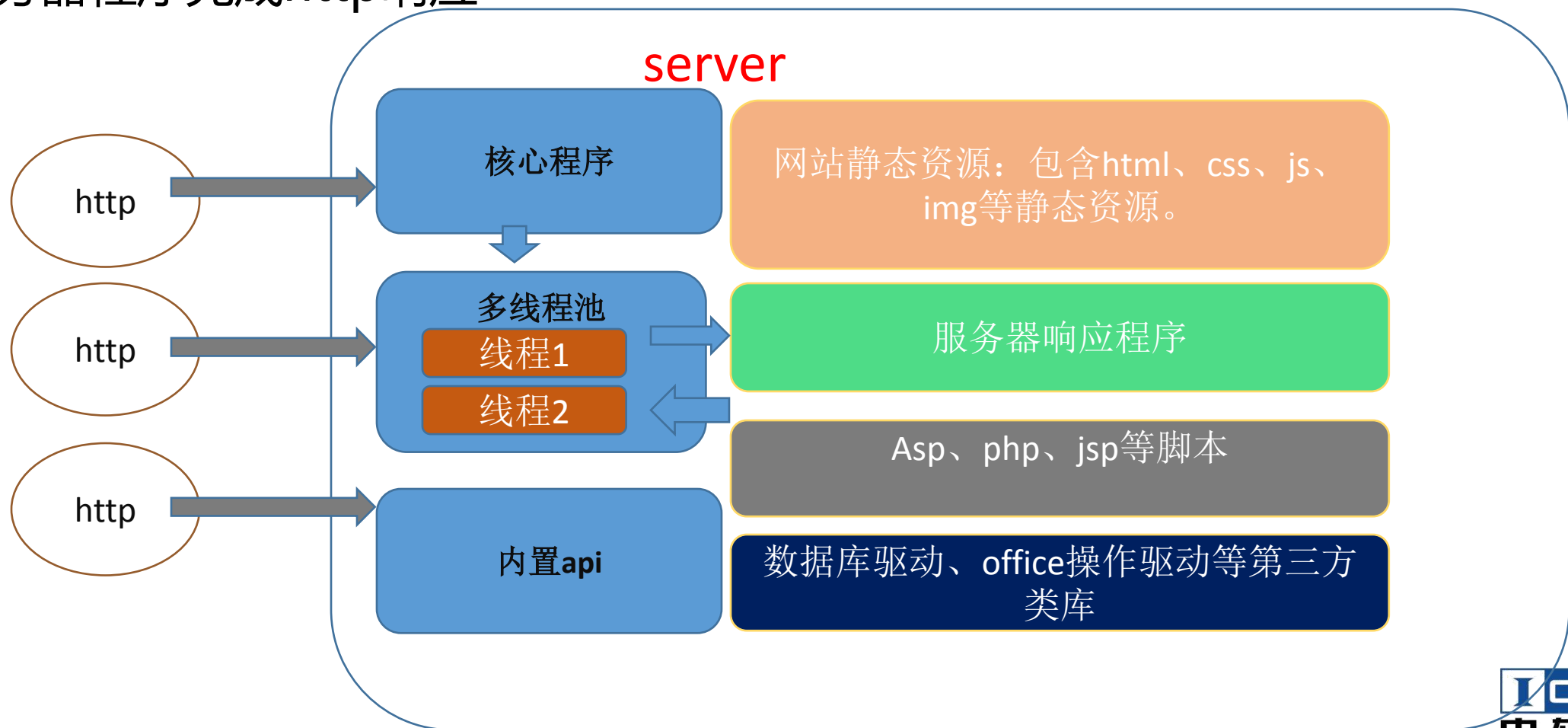
- Js程序刷新或切换网页地址

```
window.location.href = "testHTTP.html?userName=zhongruan"
```

- 通过发送AJAX发送HTTP请求 (**本节介绍**)。

知识点3- 服务器响应方式-1

- web服务器软件：一般指网站服务器安装的服务器软件，可解析http请求，并调用服务器程序完成http响应



知识点3- 服务器响应方式-2

- http响应格式

```
HTTP/1.1 200 OK
Cache-Control: private, max-age=30
Content-Type: text/html; charset=utf-8
Content-Encoding: gzip
Expires: Mon, 25 May 2009 03:20:33 GMT
Last-Modified: Mon, 25 May 2009 03:20:03 GMT
Vary: Accept-Encoding
Server: Microsoft-IIS/7.0
X-AspNet-Version: 2.0.50727
X-Powered-By: ASP.NET
Date: Mon, 25 May 2009 03:20:02 GMT
Content-Length: 12173
```

-消息体的内容（略）

知识点3- 服务器响应方式-3

• http响应格式解析

- 状态行：状态行由三个标记组成：HTTP版本、响应代码和响应描述。
 - HTTP版本：向客户端指明其可理解的最高版本。
 - 响应代码：3位的数字代码，指出请求的成功或失败，如果失败则指出原因。
 - 响应描述：为响应代码的可读性解释。 例如：HTTP/1.1 200 OK
- 响应头标：像请求头标一样，它们指出服务器的功能，标识出响应数据的细节。
- 空行：最后一个响应头标之后是一个空行，发送回车符和退行，表明服务器以下不再有头标。
- 响应数据：HTML文档和图像等，也就是HTML本身

知识点4- AJAX基本概念-1

- AJAX:Asynchronous JavaScript and XML。现在允许浏览器与服务器通信而无须刷新当前页面的技术都被叫做Ajax。
 - 发展历程：
 1. 在2005年2月由Jesse James Garrett首次提出。
 2. 在google地图中首次被应用，从而为人所知。
 3. 在目前阶段通过AJAX与服务器交互已经成为主流的开发模式。
 - 优点分析：
 1. 避免网页频繁刷新造成的不良用户体验。
 2. 减少刷新将会减少服务器的执行压力。
 3. 此种交互方式提高服务器程序的公用性。

知识点4- AJAX基本概念-2

- AJAX使用场景分析：
 - Google搜索时智能提示不刷新网页。

Ajax		
ajax教程	2,030,000	结果
ajax技术	3,610,000	结果
ajax基础教程	524,000	结果
ajax框架	550,000	结果
ajax 框架	578,000	结果
ajax.net	913,000	结果
ajax 实例	1,610,000	结果
ajax 教程	2,030,000	结果
ajax in action	3,610,000	结果
ajaxpro	84,100	结果

- 地图缩小或放大时不刷新网页。



知识点4- AJAX基本概念-3

- **AJAX的执行原理**：使用ajax与服务器交互时，发送以及接收的信息是特殊数据格式，而非整个网页程序，因此不需要刷新。

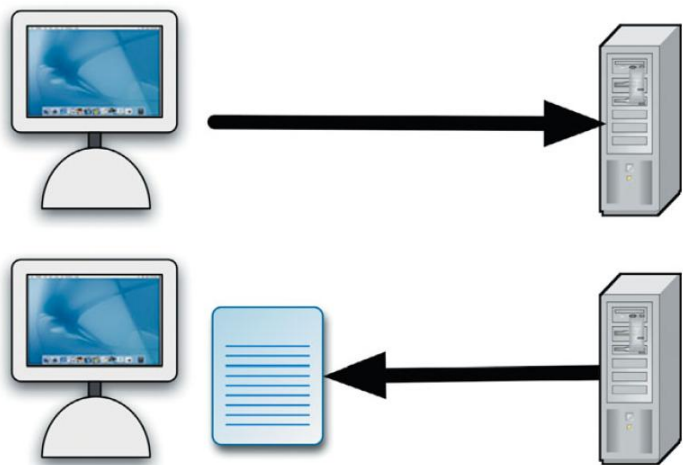


图1 Web的传统模型。客户端向服务器发送一个请求，服务器返回整个页面，如此反复

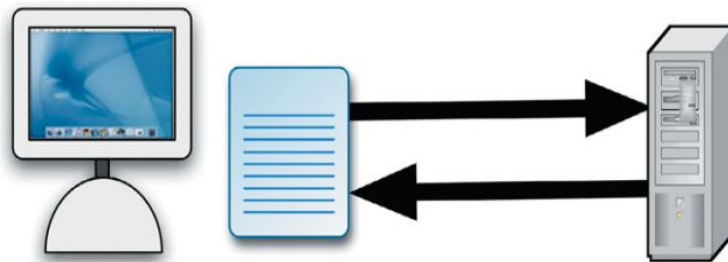


图2 在Ajax模型中，数据在客户端与服务器之间独立传输。服务器不再返回整个页面，而只返回可用信息。

知识点4- AJAX基本概念-4

- Ajax的技术组成：AJAX实际上是几种技术以一种全新的方式聚合在一起的方式。
 - **服务器端语言**：服务器响应客户端请求。 **Ajax与服务器端语言无关。**
 - **XML** (eXtensible Markup Language , 可扩展标记语言) 是一种描述数据的格式。**XML 是传输格式的其中的一种选择。**
 - **HTML**和 **CSS** (Cascading Style Sheet,级联样式单) **标准化呈现；**
 - **DOM** (Document Object Model,文档对象模型) **实现动态显示和交互；**
 - 使用XMLHTTP组件**XMLHttpRequest对象**进行**同步或异步数据读取,非w3c标准**
 - 使用**JavaScript****绑定和处理所有数据。**

知识点5- XHR对象-1

- XHR对象：是XMLHttpRequest对象的缩写，封装在window对象中，封装了发送AJAX请求、接收响应的属性以及方法。最早是在IE5中以ActiveX组件的形式实现的，现在已经被所有高级浏览器兼容。
- XHR对象的创建

```
function getXHR() {  
    if (window.XMLHttpRequest) {  
        return new window.XMLHttpRequest;  
    } else {  
        //为了解决IE低版本兼容性。  
        return new window.ActiveXObject("Microsoft.XMLHTTP");  
    }  
}
```

如果XMLHttpRequest对象存在，则把 xhr 的值设为该对象的新实例。如果不存在，就去检测 ActiveObject 的实例是否存在，如果答案是肯定的，则把微软XMLHTTP 的新实例赋给 xhr

知识点5- XHR对象-2

• XHR对象内置方法：







绿色为发送ajax请求
关键步骤

方法	描述
abort()	停止当前请求
getAllResponseHeaders()	把HTTP请求的所有响应首部作为键/值对返回
getResponseHeader(“header”)	返回指定首部的串值
open(“method”, “url”)	建立对服务器的调用。Method参数可以是GET、POST或PUT. url参数可以是相对URL或绝对URL。
send(content)	向服务器发送请求
setRequestHeader(“header”, “value”)	把指定首部设置为所提供的值。在设置任何首部之前必须先调用open()

知识点5- XHR对象-3

• XHR对象内置属性：

绿色为发送ajax请求
关键步骤

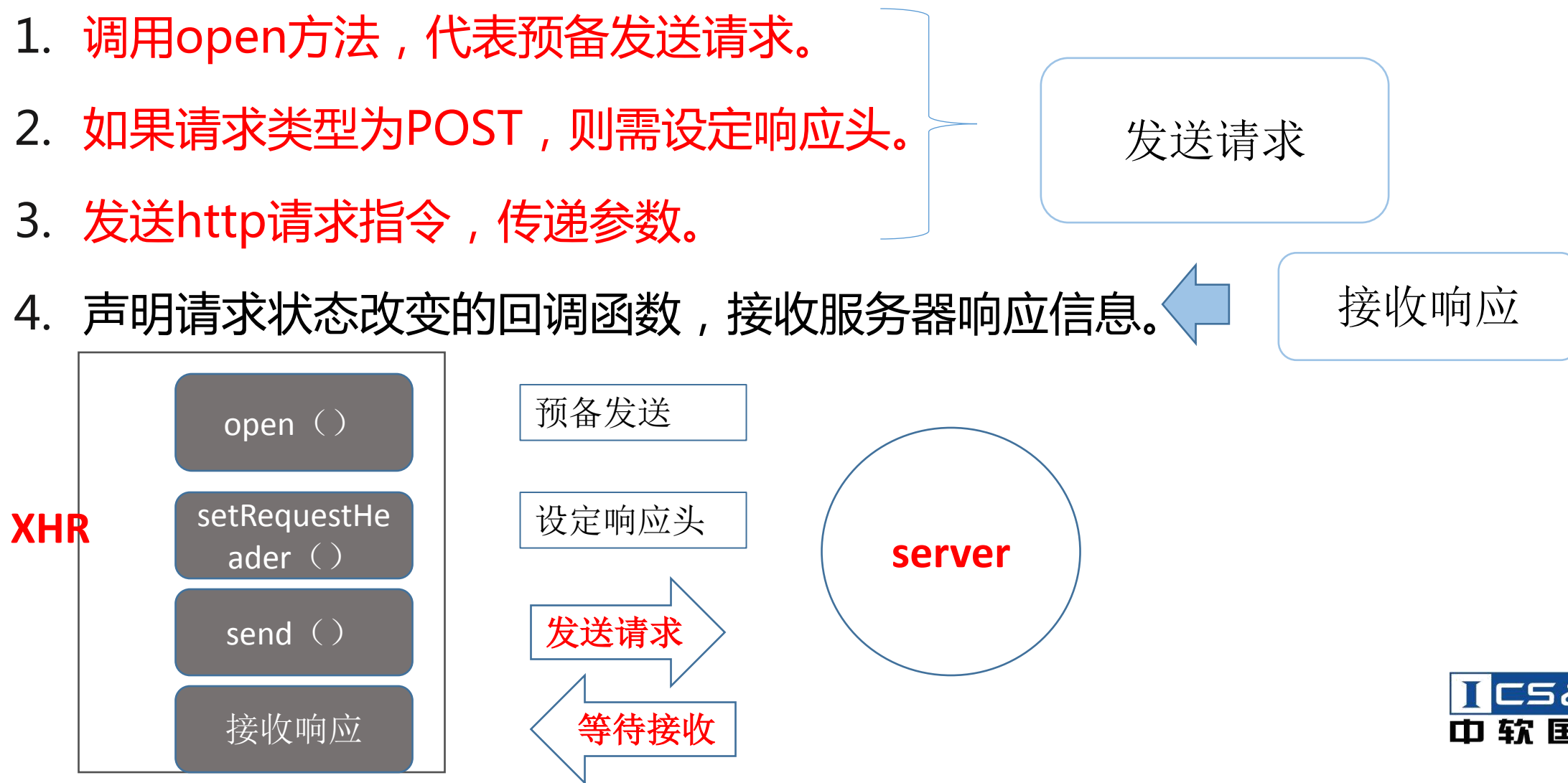
属性	描述
 onreadystatechange	每个状态改变是都会触发这个事件处理器，通常会调用一个javaScript函数
 readyState	请求的状态，有5个可取值：0=未初始化、1=正在加载、2=已经加载、3=交互中、4=完成。
 responseText	服务器的响应，表示为一个串。
 responseXML	服务器的响应，表示为XML。这个对象可以解析为DOM对象。
 status	服务器的HTTP状态码（200对应OK、404对应NotFount、等）
 statusText	HTTP状态码的相应文本（OK或NotFount等）

了解这些接口后，我们就使用这些接口发送AJAX请求。

知识点6- 发送ajax请求-1

• 发送AJAX请求的步骤：

1. 调用open方法，代表预备发送请求。
2. 如果请求类型为POST，则需设定响应头。
3. 发送http请求指令，传递参数。
4. 声明请求状态改变的回调函数，接收服务器响应信息。



知识点6- 发送ajax请求-2

- open(method, url, asynch)：预备发送请求。
 - method：请求类型，类似“GET”或“POST”的字符串，**若使用GET请求，则可以在url地址栏传递参数**，但原则上字符数量**不超过2000**，且不同服务器解码方式不同**容易造成乱码,可以使用URI编码**。
 - url：路径字符串，可以为get请求传递参数，有些浏览器会把多个请求的结果缓存在同一个URL，造成网页更新不及时，**可在url中增加时间戳避免缓存**。
 - asynch：表示请求是否要异步传输，默认值为true。同步传输将导致网页等待期间无法操作，建议使用异步传输。

```
xhr.send("GET","../../testGet?userName=user",true);
```

知识点6- 发送ajax请求-3

- `setRequestHeader(header,value)`:设定响应头，响应头用于描述元数据。
 - `header`：响应头的名字;如果用 **POST** 请求向服务器发送数据，需要将“Content-type” 的首部设置为 “**application/x-www-form-urlencoded**” . 它会告知服务器正在发送数据，并且数据已经符合URL编码了
 - `value`：响应头的值。

注意事项：**该方法必须在open()之后才能调用**

```
var xhr = getXHR();  
xhr.open("POST", ".../testPost", true);  
xhr.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
```

知识点6- 发送ajax请求-4

- `send(data)`：根据`open`方法设定的参数，发送实际http请求指令。
 - `data`：将要传递给服务器的字符串（字符类型）。
- 注意事项：
 - 若选用的是 GET 请求，则不会发送任何数据。
 - Post请求发送的数据类型为用=分开的名值对，如果需要传送JSON对象，则需要转型。（JSON格式是最常用格式，表现为一个js对象）。

```
//发送字符  
xhr.send( "userName=zhangsan&&password=123" );  
//发送json对象需转型（详见之后知识点）  
xhr.send( "param=" + JSON.stringify(obj));
```

知识点6- 发送ajax请求-5

- 完整的发送get请求

```
var xhr = getXHR();  
//加上时间戳避免缓存，尤其在生成验证码时，经常使用  
xhr.open("GET","../testGet?userName=user&&now=" +  
Date.now(),true);  
xhr.send(null);
```

- 完整的发送post请求

```
var xhr = getXHR();  
xhr.open("POST","../testPost",true);  
xhr.setRequestHeader("Content-type","application/x-www-form-urlencoded");  
var obj = {userName:'zhangsan',password:'123'};  
xhr.send("param=" + JSON.stringify(obj));
```

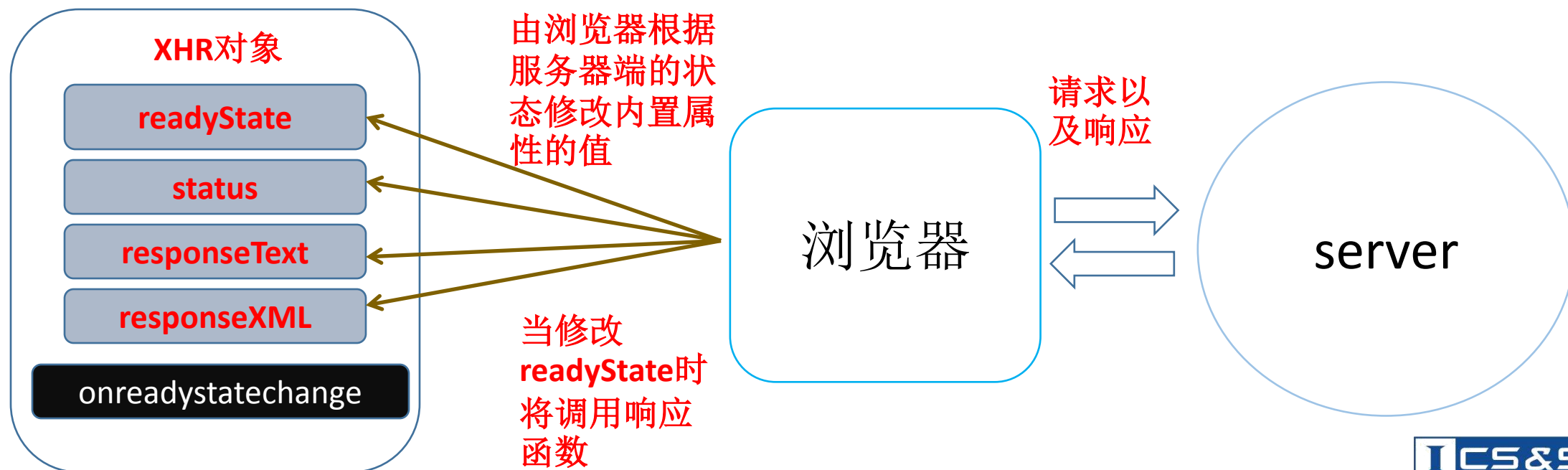
知识点7- 接收服务器响应-1

- 接收服务器响应依靠XHR的如下接口实现：
 - onreadystatechange:服务器响应状态改变的事件属性。
 - readyState：表示Ajax请求的当前状态，如发送中，请求结束等。
 - status：服务器返回的http响应状态，如200代表正常404代表没找到资源。
 - .responseText：服务器返回的字符格式。
 - responseXML：服务器返回的XML格式（前端开发中使用很少）。

xhr上述属性都是由服务器端修改，而非客户端修改，因此是只读属性。

知识点7- 接收服务器响应-2

- **onreadystatechange**:服务器响应状态改变的事件属性，值为函数类型。服务器会通知客户端当前的通信状态。
 - XMLHttpRequest 对象的 **readyState**是由浏览器更新。
 - 每次 **readyState** 属性的改变都会触发 **readystatechange** 事件



知识点7- 接收服务器响应-3

- readyState：表示Ajax请求的当前状态,取值范围如下：
 - 0 代表未初始化。还没有调用 open 方法
 - 1 代表正在加载。 open 方法已被调用，但 send 方法还没有被调用
 - 2 代表已加载完毕。 send 已被调用。请求已经开始
 - 3 代表交互中。
 - 4 代表完成。响应发送完毕。（一般情况下只需判断此状态即可）。

编写时应注意把onreadystatechange赋值编写在open之前。

知识点7- 接收服务器响应-4

- status：服务器发送的响应状态码，常用状态码及其含义：
 - 404 没找到页面(not found)
 - 403 禁止访问(forbidden)
 - 500 内部服务器出错(internal service error)
 - 200 一切正常(ok)
 - 304 没有被修改(not modified)

注意事项：通过XHR.status和200比较，可以确保服务器是否已发送了一个成功的响应

知识点7- 接收服务器响应-5

- **responseText**:服务器返回的响应数据，字符类型，**建议使用**。
 - XMLHttpRequest 的 **responseText** 属性包含了从服务器发送的数据。**它是一个HTML,XML或普通文本**，这取决于服务器发送的内容。
 - 当 **readyState 属性值变成 4 时**, **responseText** 属性才可用，表明 Ajax 请求已经结束。
- **responseXML**：服务器返回的响应数据，格式为xml，字符类型。
 - 如果服务器返回的是 XML，那么数据将储存在 **responseXML** 属性中。
 - 只用服务器**发送了带有正确头部信息的数据时**，**responseXML** 属性才是可用的。**MIME 类型必须为 text/xml**

知识点7- 接收服务器响应-6

- 完整的接收服务器响应程序：

```
xhr.onreadystatechange = function(e) {  
    if(e.target.readyState == 4 ) {  
        if (e.target.status == 200 || e.target.status == 304) {  
            alert(e.target.responseText);  
        }  
    }  
}
```

知识点8- 传输格式-1

- AJAX 是一门与语言无关的技术，例如网页、android、IOS，都可以使用AJAX，同时多种web服务器都可以响应AJAX。

此模式要求不同客户端的ajax请求与响应的数据格式相同。



Android



前后端交互的数据格式

1 xml

2 html

3 json（最常用）



- 多客户端都可以统一访问一个服务器
- 不同语言开发的服务器端程序都可以被不同客户端访问。

知识点8- 传输格式-2

- JSON (JavaScript Object Notation) : 一种简单的数据格式, 比xml更轻巧。
 - JSON是JavaScript原生格式, 这意味着在JavaScript中处理JSON数据不需要任何特殊的API或工具包。 `{"user":"zhongruan"}`
 - JSON的基本规则就是无序的名值对, 但严格意义上名称要用" " 包裹。数据值只能为数字、字符、数组、布尔、JSON对象。

`{ "flag" :true} { "fav" :["java" ," js"]}`

- JSON对象中多个属性名用逗号分开 `{"key1":"val1","key2":"val2"}`
- JSON对象存在嵌套关系, 可以无限嵌套 `{"key1":{"key11":{"key111":"val11"}}`

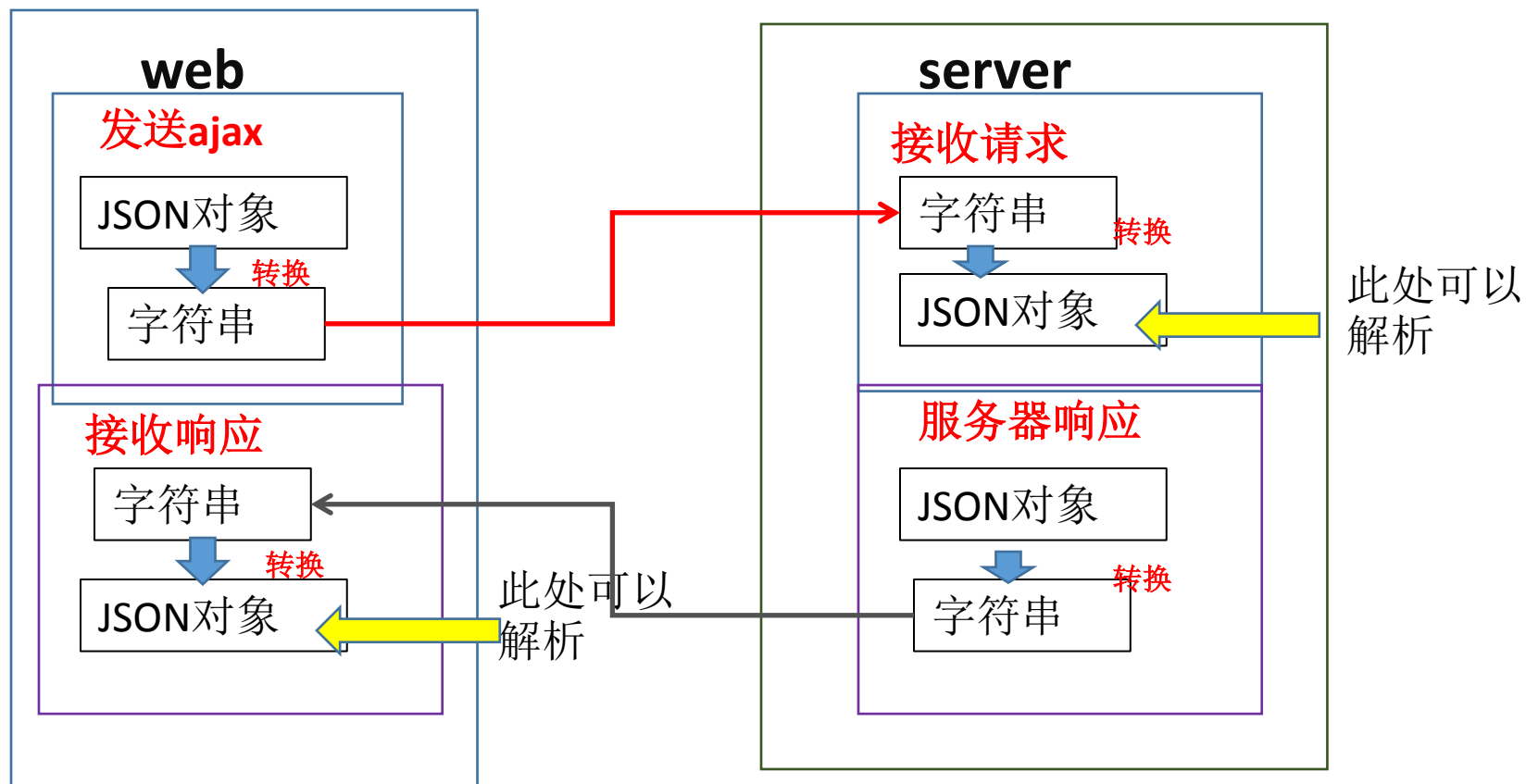
知识点8- 传输格式-3

- JSON格式的解析：JSON是一种数据格式，在不同语言中解析都需要用不同的类库，**但在js中就是原生语法，使用属性名访问属性值，使用下标访问数组。**
 - 代码示例

```
var a = {"key1":{"key11":{"key111":"val11"}}}  
alert(a.key1.key11.ke111);//运行结果val11  
var z = {"key1":[1,2,3,{"key11":"val11"}]};  
alert(z.key1[3].key11);//val11
```


知识点8- 传输格式-4

• JSON格式的转换



知识点8- 传输格式-5

- ECMAScript中扩展了字符与JSON格式相互转换的方式。

`String JSON.stringify(obj)` : 将ECMAScript对象转换为字符。

--String : 返回类型。

--JSON : window对象的全局属性，但只有在高版本浏览器中兼容。

--stringify:JSON对象的全局方法，将目标对象转换为字符。

--obj:需要转换的对象。

```
var obj = {userName:'zhangsan', password:'123'};  
xhr.send("param=" + JSON.stringify(obj));
```

知识点8- 传输格式-6

- **Obj** **JSON.parse(Str)** : 将JSON格式的字符转换为ECMAScript对象。

--obj : 返回类型 , 对象。

--str:需要转换的字符。

```
var str = ' {"user": "zhongruan"} ' ;  
alert (JSON.parse(str).user);
```

兼容性解决方案：JSON提供了json.js包，下载<http://www.json.org/json.js>后，使用parseJSON()方法将字符串解析成 JS 对象

知识点9- JSON文件

- json文件：以json为后缀，内容为标准json格式的文件，用于模拟服务器返回数据，但只能以get方式发送请求。
 - Json文件可以保证前端人员与后端人员同时开发。
 - Json文件可以用于前端开发的信息配置。
 - Json文件中的字符类型取值需要使用双引号包裹。

```
{"userName":"zhangsan",  
"petName":"老张"}
```

第2节 【jquery框架】

- 知识点1：jquery简介
- 知识点2：入门程序
- 知识点3：基本语法与核心函数
- 知识点4：ajax封装

知识点1- jquery简介-1

- 什么是jQuery：JavaScript+查询（Query），是由javascript编写的js库，为编写javascript提供了更高效便捷的接口。
- jQuery的特点：
 - jQuery是一个轻量级的javascript框架，核心理念是WRITE LESS,DO MORE。
 - jQuery应用广泛，在世界前10000个访问最多的网站中有55%在使用jQuery。
 - jQuery是免费、开源的，文档丰富。
 - 众多第三方的js库都采用jQuery辅助编写，如jquery-mobile、easyUI、jqueryUI等等。

知识点1- jquery简介-2

- jQuery的核心功能如下：

- HTML 元素选取：提供了在文档上查找dom元素的方式。
- HTML 元素操作：提供了对dom元素的操作增删改功能。
- CSS 操作：提供了对dom元素样式的修改功能。
- HTML 事件函数：扩展了javascript事件，并实现了兼容。
- JavaScript 特效和动画。
- AJAX：简化了javascript对于ajax的调用。
- Utilities：提供了若干工具函数。

知识点1- jquery简介-3

- jQuery的版本：目前jQuery已经升级到3.0+版本，但仍处于（ beta ）公测阶段。
本次课程的版本jquery1.9

- jQuery下载地址（ 可下载各种版本的jQuery ）

<http://jquery.com/download/>

- 下载文件说明：

一般建议在开发阶段使用jQuery.js,上线后修改为min文件，从而减少客户端下载量。

- jquery-1.9.1.js：jquery代码开发库，可供阅读，也可以通过断点调试程序。
- jquery-1.9.1.minjs：jquery代码迷你库，但已经删除掉所有缩进格式。（参见js目录，已经下载）。

知识点2-入门程序-1

- jQuery的引入方式：

- 加载本地的jQuery文件

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

- 引入网络上的jQuery文件

```
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.0/jquery.min.js"></script>
```

建议使用第一种引入方式，避免js服务器更换地址导致的程序错误

知识点2-入门程序-2

- 程序描述：当网页加载后弹出helloworld对话框。

```
$(document).ready(function() {  
    alert('helloworld');  
});
```

- \$.jQuery的核心函数，此处用于获取document对象。
- ready():jQuery提供的事件函数，此处代表当文档加载后调用响应函数。

知识点3-基本语法与核心函数-1

- jQuery的基本语法如下

```
$(selector).action()
```

- \$: jQuery的核心函数,也可编写成jQuery。
- selector : jQuery的查找网页元素的表达式,返回jQuery封装的DOM对象。
- action : jQuery的内置函数。

例如以下代码实现网页全部段落标签的隐藏功能。

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

知识点3-基本语法与核心函数-2

- jQuery：是jQuery的核心函数，可以缩写为\$，主要功能如下
 - 获取DOM对象：获取文档树上dom对象(jQuery的DOM对象)。
 - 创建元素：创建html元素并插入到DOM树上。
 - 克隆对象：创建一个jQuery对象的副本。
- 源码解析：jQuery函数，第一个参数为jQuery的选择器，第二个参数为DOM对象，jQuery会判断传递参数的类型。

```
jQuery = function( selector, context ) {  
    // The jQuery object is actually just the init constructor 'e'  
    return new jQuery.fn.init( selector, context, rootjQuery );  
},
```

```
9579 window.jQuery = window.$ = jQuery;
```

知识点1- ajax封装-1

- jQuery封装了多个用于发送ajax请求的底层函数，我们这里只介绍轻量好用的几个。

方法	描述
\$.get()	发送get请求，并接收服务器端返回数据
\$.getJSON()	发送get请求，并接收服务器端返回的json数据
\$.post()	发送post请求，并接受服务器端返回的json数据
obj.load()	从指定地址读取html代码并写入到指定对象中

知识点1- ajax封装-2

- 参数以及响应函数说明：以上方法参数与响应函数使用方式一致，以下以get为例介绍参数以及响应函数。
 - `get("url" ,data,function(sData,status,xhr),type):`
 - url:访问地址,字符类型
 - data：发送给服务器的数据。
 - function：服务器成功响应的回调参数
 - --sData：服务器返回数据，格式根据type决定。
 - --stats：服务器返回状态，默认为success
 - --xhr：发送请求xmlHttpRequest对象
 - type：请求类型（xml, json, script, 或 html）

