



# 第九章

## jQuery中的Ajax

# 预习检查



## 集中测试

- ◆ 简述Ajax请求与普通请求的主要区别
- ◆ Ajax主要包括哪些技术?
- ◆ 使用原生JavaScript发送Ajax请求的步骤有哪些?
- ◆ 在JavaScript中按照JSON格式定义一个新闻数组, 新闻属性包含ID、标题、发布时间、作者

# 本章任务

- ◆ 实现无刷新邮箱验证
- ◆ 验证注册邮箱是否可用



[注册](#) | [登录](#) | [帮助](#)

## 注册京东会员

您的Email:

输入密码:

再输入一遍密码:

[注册](#)

[关于我们](#) | [诚聘英才](#) | [联系方式](#) | [帮助中心](#)



再输入一遍密码:

[注册](#)

[关于我们](#) | [诚聘英才](#) | [联系方式](#) | [帮助中心](#)

# 本章目标

- ◆ 理解Ajax技术
- ◆ 掌握jQuery的\$.ajax()方法
- ◆ 掌握JSON的使用

重点

难点

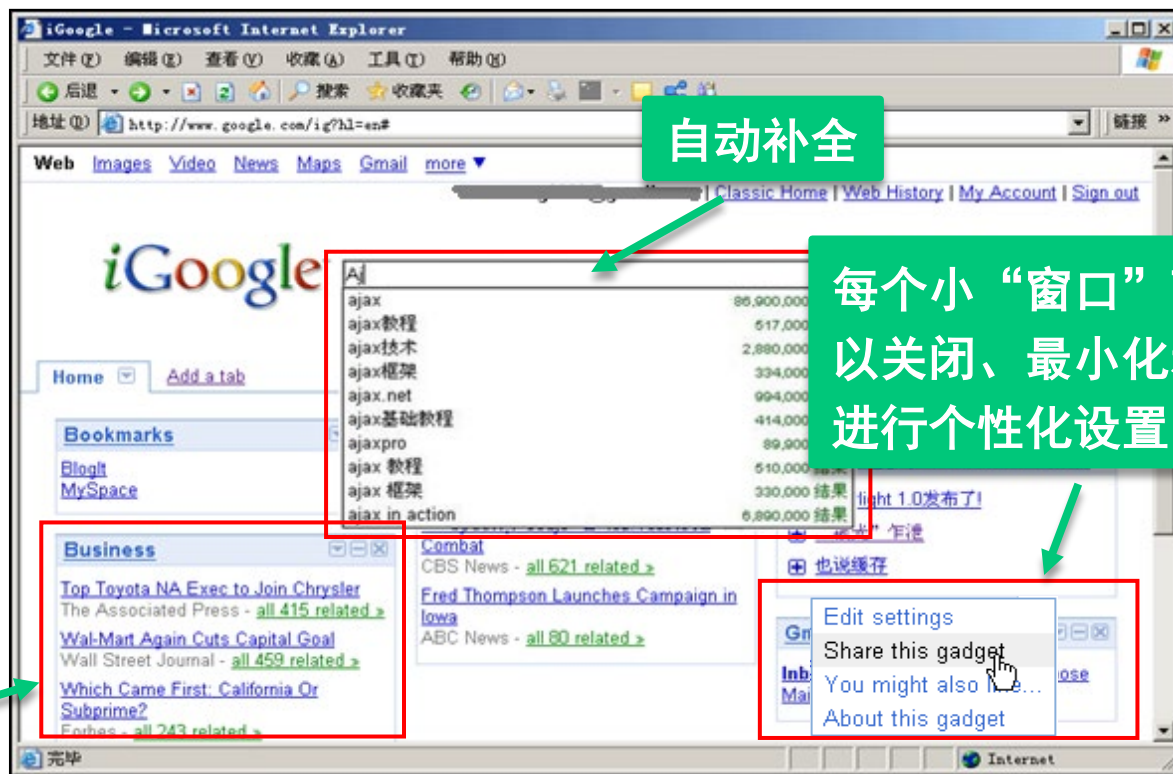
重点

难点

重点

# 认识Ajax

- ◆ Web 2.0的特点
  - ◆ 用户贡献内容
  - ◆ 内容聚合RSS
  - ◆ 更丰富的“用户体验”



!

所有操作都是在**不刷新窗口**的情况下完成的

# 为什么使用Ajax

- ◆ 无刷新：不刷新整个页面
- ◆ 无刷新的好处
  - ◆ 只更新部分页面，有效
  - ◆ 提供连续的用户体验
  - ◆ 提供类似C/S的交互效果，操作更方便



# 传统Web与Ajax的差异

差异	方式	说明
发送请求方式不同	传统Web	浏览器发送同步请求
	Ajax技术	异步引擎对象发送请求
服务器响应不同	传统Web	响应内容是一个完整页面
	Ajax技术	响应内容只是需要的数据
客户端处理方式不同	传统Web	需等待服务器响应完成并重新加载整个页面后用户才能进行操作
	Ajax技术	可以动态更新页面中的部分内容不影响用户在页面进行其他操作

# Ajax简介

- ◆ Ajax: 异步刷新技术

The logo for AJAX features the word "AJAX" in a bold, blue, sans-serif font. A large, blue, double-headed arrow is positioned behind the letters "A" and "J", pointing both up and down.

**A**ynchronous  
异步的

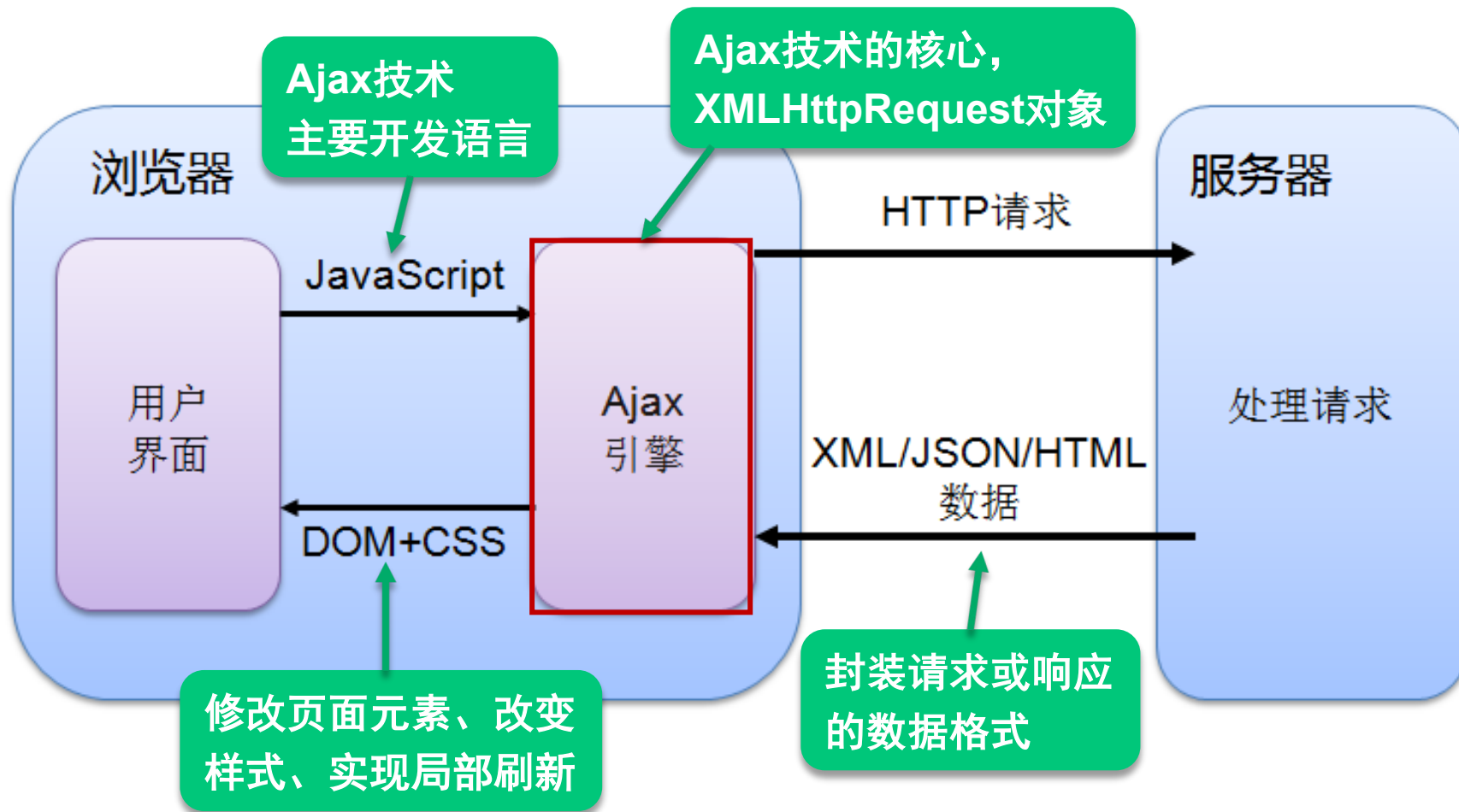
**J**avaScript

**A**nd

**X**ml



# Ajax工作流程



# XMLHttpRequest 3-1

- ◆ 整个Ajax技术的核心
- ◆ 提供异步发送请求的能力

- ◆ 常用方法

方 法	说 明
open( String method, String url, boolean async, String user, String password )	创建一个新的HTTP请求
send( String data )	发送请求到服务器端
abort( )	取消当前请求
setRequestHeader( String header, String value )	设置请求的某个HTTP头信息
getResponseHeader(String header)	获取响应的指定HTTP头信息
getAllResponseHeader( )	获取响应的所有HTTP头信息

# XMLHttpRequest 3-2

## ◆ 事件

◆ onreadystatechange: 指定回调函数

## ◆ 常用属性

◆ readyState: XMLHttpRequest的状态信息

就绪状态码	说 明
0	XMLHttpRequest对象未完成初始化
1	XMLHttpRequest对象开始发送请求
2	XMLHttpRequest对象的请求发送完成
3	XMLHttpRequest对象开始读取响应
4	XMLHttpRequest对象读取响应结束

# XMLHttpRequest 3-3

## ◆ 常用属性 (续)

### ◆ status: HTTP的状态码

状态码	说 明
200	服务器正确返回响应
404	请求的资源不存在
500	服务器内部错误
403	没有访问权限
...	.....

! 就绪状态是4且状态码是200, 表示正确完成

### ◆ statusText: 返回当前请求的响应状态

### ◆ responseText: 以文本形式获得响应的内容

### ◆ responseXML: 将XML格式的响应内容解析成 DOM对象返回

# 使用Ajax验证用户名2-1



问题

## ◆ 实现无刷新用户名验证

- ◆ 当用户名文本框失去焦点时，发送请求到服务器，判断用户名是否存在
- ◆ 如果已经存在提示“用户名已被使用”，如果不存在则提示“用户名可以使用”

用户名:  用户名可以使用!

用户名:  用户名已被使用!

# 使用Ajax验证用户名2-2



分析

- ◆ 使用文本框的onBlur事件
- ◆ 使用Ajax技术实现异步交互
  - ◆ 创建XMLHttpRequest对象
  - ◆ 通过XMLHttpRequest对象设置请求信息
  - ◆ 向服务器发送请求
  - ◆ 创建回调函数，根据响应状态动态更新页面



演示示例1：原生Ajax请求

# GET请求和POST请求的区别

步骤	请求方式	实现代码
初始化组件	GET	<code>xmlHttpRequest.open( "GET", url, true );</code>
	POST	<code>xmlHttpRequest.open( "POST", url, true );</code> <code>xmlHttpRequest</code> <code>.setRequestHeader( "Content-Type",</code> <code>"application/x-www-form-urlencoded" );</code>
发送请求	GET	<code>xmlHttpRequest.send( null );</code>
	POST	<code>xmlHttpRequest.send(</code> <code>"key=xxx&amp;type=12&amp;year=2016" );</code>

# 学员操作——实现无刷新邮箱验证



练习

## ◆ 需求说明

### ◆ 实现无刷新验证注册邮箱

### ◆ 在用户注册页面，当E-mail文本框失去焦点时判断邮箱是否已被使用(请求URL: /resp.asp?email=\*\*) )

您的Email:

用户名:

输入密码:

再输入一遍密码:

您的Email:

该邮箱已被注册!

用户名:

输入密码:

再输入一遍密码:


完成时间：20分钟





# 共性问题集中讲解

- ◆ 常见问题及解决办法
- ◆ 代码规范问题
- ◆ 调试技巧



共性问题集中讲解

# 使用jQuery实现Ajax



问题

## ■ 传统方式实现Ajax的不足

- ◆ 步骤繁琐
- ◆ 方法、属性、常用值较多不好记忆
- ◆ 处理复杂结构的响应数据（如XML格式）比较烦琐
- ◆ 浏览器兼容问题



jQuery将Ajax相关操作都进行了封装

# jQuery中的Ajax方法

方法名称	说明
<code>\$.get()</code>	通过HTTP GET请求从服务器加载数据
<code>\$.post()</code>	通过HTTP POST请求从服务器加载数据
<code>load()</code>	从服务器加载数据，并把返回的HTML插入到匹配的DOM元素中
<code>\$.ajax()</code>	执行一个异步HTTP（Ajax）请求
<code>\$.getJSON()</code>	通过HTTP GET请求从服务器加载JSON编码格式的数据
<code>\$.getScript()</code>	通过HTTP GET请求从服务器加载JavaScript文件并执行该文件

# \$.get()方法与\$.post()方法

## 语法

```
$.get(url,data,success(response,status,xhr),dataType); // $.get()语法  
$.post( url, data, success(response,status,xhr),datyType);// $.post()语法
```

## 示例

```
$(function(){ // 使用$.get()方法获取test.txt中的文本信息
```

```
$("#btn1").click(function(){
```

```
$.get("test.txt",function(data){
```

```
$("#showTestMsg").html(data);
```

```
},"text");
```

```
success(response,sta
```

```
tus,xhr)
```

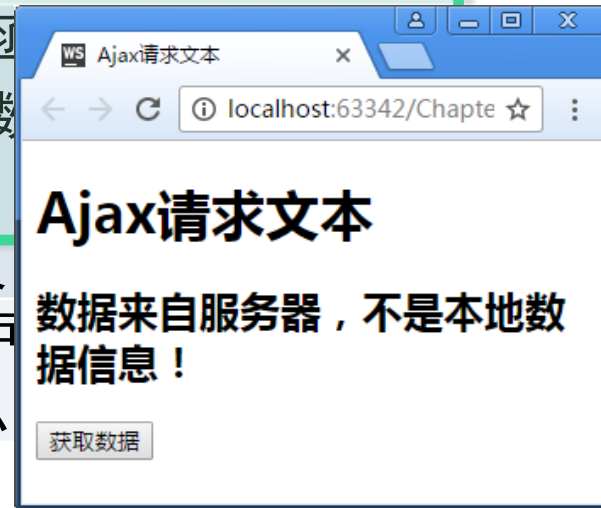
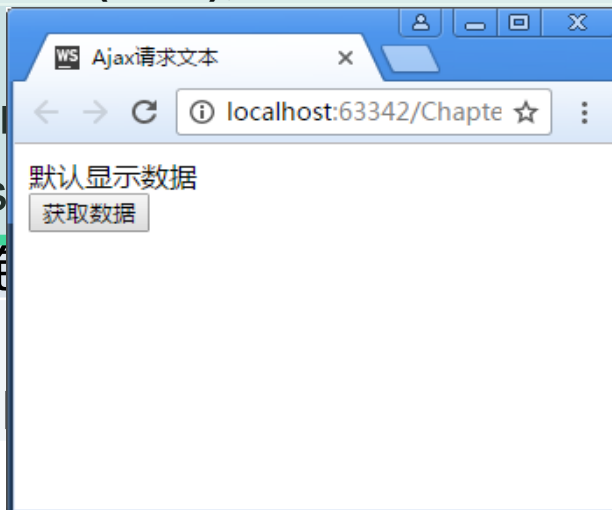
必需。规定将请求发送给的URL地址。

可选。规定连同请求发送到服务器的数据。

可选。

response  
status

```
ex.html • test.txt ×  
<h1>Ajax请求文本</h1>  
<h2>数据来自服务器，不是本地数据信息! </h2>
```



演示示例02: jQuery \$.get

# jQuery中的load()方法

## 语法

```
$(selector).load(url,data,function(result,status, xhr));
```

参数	说明
String url	必选，规定将请求发送到哪个URL
Object data或String data	可选，规定连同请求发送到服务器的数据
function callback (Object result, String status, XMLHttpRequest xhr)	可选，请求完成后调用的函数 参数result：来自请求的结果数据 参数status：请求的状态 参数xhr：XMLHttpRequest对象



演示示例02: jQuery \$.get

# \$.ajax()简介2-1

语法

```
$.ajax( [settings] );
```

## 常用属性参数

参数	类型	说明
url	String	发送请求的地址，默认为当前页地址
type	String	请求方式，默认为GET
data	PlainObject 或 String 或 Array	发送到服务器的数据
dataType	String	预期服务器返回的数据类型，包括：XML、HTML、Script、JSON、JSONP、text
timeout	Number	设置请求超时时间
global	Boolean	表示是否触发全局Ajax事件，默认为true

# \$.ajax()简介2-2

## ◆ 常用函数参数

参 数	类 型	说 明
beforeSend	Function ( jqXHR jqxhr, PlainObject settings )	发送请求前调用的函数
success	Function( 任意类型 result, String textStatus, jqXHR jqxhr )	请求成功后调用的函数 参数result: 可选, 由服务器返回的数据
error	Function ( jqXHR jqxhr, String textStatus, String errorThrown )	请求失败时调用的函数
complete	Function ( jqXHR jqxhr, String textStatus )	请求完成后 (无论成功还是失败) 调用的函数

# 使用\$.ajax()发送异步请求2-1

 问题 如何使用\$.ajax()方法实现无刷新用户名验证?

用户名:  用户名可以使用!

用户名:  用户名已被使用!



# 使用\$.ajax()发送异步请求2-2

## 示例

```
$.ajax( {  
  "url"      : "url",           // 要提交的URL路径  
  "type"     : "get",          // 发送请求的方式  
  "data"     : data,           // 要发送到服务器的数据  
  "dataType" : "text",         // 指定传输的数据格式  
  "success"  : function(result) { // 请求成功后要执行的代码  
    },  
  "error"    : function() {     // 请求失败后要执行的代码  
    }  
});
```



演示示例03：名称验证

# 学员操作——验证注册邮箱是否可用

## ◆ 需求说明

### ◆ 使用\$.ajax()实现无刷新邮箱验证



The screenshot shows the JD.com registration page. At the top left is the JD.com logo (a white dog head) and the text "京东". At the top right are links for "注册 | 登录 | 帮助". The main content area is titled "注册京东会员" and contains three input fields: "您的Email:", "输入密码:", and "再输入一遍密码:". Below these fields is a red "注册" button. At the bottom of the page are links for "关于我们 | 诚聘英才 | 联系方式 | 帮助中心".




The screenshot shows the JD.com registration page after an email validation error. The "您的Email:" field now contains the text "102211267@qq.com". Below this field, the text "此邮箱号已注册!" is displayed in red. The "输入密码:" and "再输入一遍密码:" fields are filled with ".....". The red "注册" button remains visible. The top and bottom navigation elements are identical to the previous screenshot.

完成时间：15分钟



# 共性问题集中讲解

- ◆ 常见问题及解决办法
- ◆ 代码规范问题
- ◆ 调试技巧



共性问题集中讲解

# 认识JSON

## ◆ JSON (JavaScript Object Notation)

- ◆ 一种轻量级的数据交换格式
- ◆ 采用独立于语言的文本格式
- ◆ 通常用于在客户端和服务端之间传递数据

## ■ JSON的优点

- ◆ 轻量级交互语言、结构简单、易于解析

# 定义JSON2-1

## ◆ 定义JSON对象

语法

```
var JSON对象 = { "name" : value, "name" : value, ..... };
```

示例

```
var person = { "name" : "张三", "age" : 30, "spouse" : null };
```

# 定义JSON2-2

## ◆ 定义JSON数组

☰ 语法

```
var JSON数组 = [ value, value, ..... ];
```

☰ 示例

```
var countryArray = [ "中国", "美国", "俄罗斯" ];  
var personArray = [ { "name":"张三", "age":30 },  
                    { "name":"李四", "age":40 } ];
```

# 读取和展示JSON数据

## ◆ 定义JSON格式数据并在页面输出

一、JSON格式的用户对象:

ID: 1

用户名: 张三

密码: 000

二、JSON格式的字符串数组:

中美俄

三、JSON格式的用户对象数组:

ID 用户名 密码

2 admin 123

3 詹姆斯 11111

4 梅西 6666



演示示例04: JSON数据

# \$.getJSON()和\$.getScript ()方法2-1

 问题 JSON数据量过大时如何获取数据?

## ◆ \$.getJSON()

 语法

```
$.getJSON(url,data,success(result,status, xhr))
```

 示例

```
$.getJSON("test.json",function(data){  
    $("#showTestMsg").html(data.firstname+" "+data.lastname+" "+data.old);  
},"text");
```



演示示例05: \$.getJSON



# \$.getJSON()和\$.getScript ()方法2-2

## ◆ \$.getScript ()

☰ 语法

```
$.getScript (url, callback())
```

☰ 示例

```
$.getScript("test.js",function(){  
    $("#showTestMsg").html(data.firstname+" "+data.lastname+" "+data.old);  
},"text");
```



演示示例05: \$.getJSON

# 学员操作——制作京东页面轮播图片效果



练习

## ◆ 需求说明

- ◆ 定义JSON格式的数据信息存储到json.js文件中
- ◆ 使用\$.ajax()方法获取json.js文件中的数据并渲染到界面上




完成时间：20分钟



# 共性问题集中讲解

- ◆ 常见问题及解决办法
- ◆ 代码规范问题
- ◆ 调试技巧



共性问题集中讲解

# 总结2-1

## jQuery中的Ajax

### 异步请求机制

#### 与传统请求的区别

请求发送方式方面  
服务器响应内容方面  
页面的处理流程方面

#### 异步请求的核心

##### 方法

open  
send

##### 事件 - onreadystatechange

##### 属性

readyState  
status  
responseText  
responseXML  
.....

#### 主要技术

**XMLHttpRequest**  
JavaScript  
DOM + CSS  
XML/**JSON**/XHTML等

#### 实现步骤

创建XMLHttpRequest对象  
初始化XMLHttpRequest组件  
设置回调函数

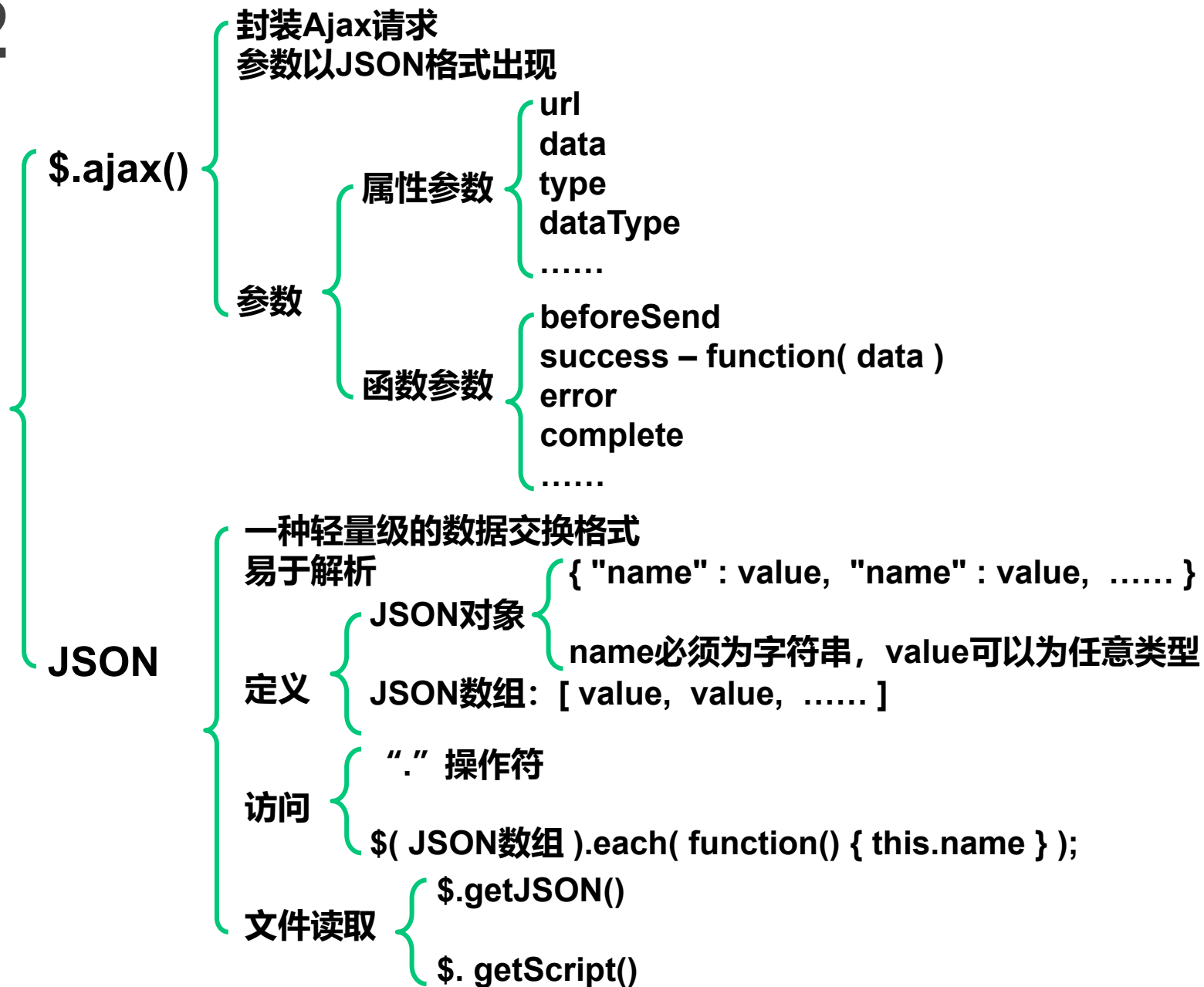
#### 发送请求

GET - `xmlHttpRequest.send( null );`

#### POST

```
xmlHttpRequest  
.setRequestHeader( ... );  
  
xmlHttpRequest.send(  
"key=xxx&type=12&..." );
```

# 总结2-2



# 作业

## ◆ 课后作业

- ◆ 教员备课时根据班级情况在此添加内容，应区分必做、选做内容，以满足不同层次学员的需求

## ◆ 预习作业

- ◆ 教员备课时根据班级情况在此添加预习内容