

## 1:什么是ajax? ajax作用是什么?

异步的javascript和xml **AJAX** 是一种用于创建快速动态网页的技术。 ajax用来与后台交互

## 2:原生js ajax请求有几个步骤? 分别是什么

```
//创建 XMLHttpRequest 对象
var ajax = new XMLHttpRequest();
    //规定请求的类型、URL 以及是否异步处理请求。
ajax.open('GET',url,true);
    //发送信息至服务器时内容编码类型
ajax.setRequestHeader("Content-type", "application/x-www-form-
urlencoded");
    //发送请求
ajax.send(null);
    //接受服务器响应数据
ajax.onreadystatechange = function () {
    if (obj.readyState == 4 && (obj.status == 200 || obj.status ==
304)) {
        }
    };
};
```

## 3:json字符串转换集json对象、json对象转换json字符串

```
//字符串转对象
JSON.parse(json)
eval('(' + jsonstr + ')')
// 对象转字符串
JSON.stringify(json)
```

## 4:ajax几种请求方式? 他们的优缺点?

2种 post,get

- 1: **get**数据是在url上, **post**把提交的数据放置在是HTTP包的body体中.
- 2: **post**请求的安全性要比**get**请求高
- 3: **get**是获取数据, **post**是提交数据
- 4: **get**传送数据只有1k, **post**理论没有限制

## 5:什么情况造成跨域?

同源策略限制 不同源会造成跨域

<http://www.baidu.com/8080/index.html>

http://	协议不同
www	子域名不同
<a href="http://www.baidu.com">baidu.com</a>	主域名不同
8080	端口号不同
<a href="http://www.baidu.com">www.baidu.com</a>	ip地址和网址不同

## 6:跨域解决方案有哪些?

### 1:jsonp 只能解决get跨域

- 原理: `<script>`是可以跨域的, 而且在跨域脚本中可以直接回调当前脚本的函数。
- 限制: 需要创建一个DOM对象并且添加到DOM树, 只能用于GET方法

```
// URL: http://a.com/foo
var callback = function(data){
    // 处理跨域请求得到的数据
};
var script = $('<script>', {src: 'http://b.a.com/bar'});
$('body').append(script);
```

---

## 2:CORS：跨域资源共享

- 原理：服务器设置Access-Control-Allow-OriginHTTP响应头之后，浏览器将会允许跨域请求
- 限制：浏览器需要支持HTML5，可以支持POST，PUT等方法兼容ie9以上

需要后台设置

```
Access-Control-Allow-Origin: *           //允许所有域名访问，或者
Access-Control-Allow-Origin: http://a.com //只允许所有域名访问
```

## 3:设置 document.domain

- 原理：相同主域名不同子域名下的页面，可以设置document.domain让它们同域
- 限制：同域document提供的是页面间的互操作，需要载入iframe页面

```
// URL http://a.com/foo
var ifr = document.createElement('iframe');
ifr.src = 'http://b.a.com/bar';
ifr.onload = function(){
    var ifrdoc = ifr.contentDocument || ifr.contentWindow.document;
    ifrdoc.getElementById("foo").innerHTML);
};

ifr.style.display = 'none';
document.body.appendChild(ifr);
```

## 4:用Apache做转发（逆向代理），让跨域变成同域

## 7:jsonp实现原理

动态创建一个srcipt标签 把接口绑定到src属性上面 利用srcipt的src属性不受同源策略限制

# 8:http常见状态码有哪些？

---

## 一: 2开头状态码

2xx (成功)表示成功处理了请求的状态代码

200 (成功) 服务器已成功处理了请求。 通常。

## 二: 3开头状态码

3xx (重定向) 表示要完成请求，需要进一步操作。 通常，这些状态代码用来重定向。

304 (未修改) 自从上次请求后，请求的网页未修改过。 服务器返回此响应时，不会返回网页内容。

## 三: 4开头状态码

4xx(请求错误) 这些状态代码表示请求可能出错，妨碍了服务器的处理

1:400 (错误请求) 服务器不理解请求的语法。

2:403 (禁止) 服务器拒绝请求。

3:404 (未找到) 服务器找不到请求的网页。

## 四: 5开头状态码

5xx(服务器错误)这些状态代码表示服务器在尝试处理请求时发生内部错误。 这些错误可能是服务器本身的错误，而不是请求出错

500 (服务器内部错误) 服务器遇到错误，无法完成请求。

501 (尚未实施) 服务器不具备完成请求的功能。 例如，服务器无法识别请求方法时可能会返回此代码。

**502**（错误网关）服务器作为网关或代理，从上游服务器收到无效响应。

**503**（服务不可用）服务器目前无法使用(由于超载或停机维护)。通常，这只是暂时状态。

**504**（网关超时）服务器作为网关或代理，但是没有及时从上游服务器收到请求。

**505**（HTTP 版本不受支持）服务器不支持请求中所用的 HTTP 协议版本。