### ajax返回的状态（单向的过程）（readyState）

0 － （未初始化）还没有调用send()方法

1 － （载入）已调用send()方法，正在发送请求

2 － （载入完成）send()方法执行完成，已经接收到全部响应内容

3 － （交互）正在解析响应内容

4 － （完成）响应内容解析完成，可以在客户端调用了

### 实现一个Ajax（创建、open、send、onreadystatechange、readyState&&status、responseText）7点

AJAX创建异步对象XMLHttpRequest，操作XMLHttpRequest 对象

（1）设置请求参数（请求方式（get），请求页面的相对路径（url），是否异步）

（2）设置回调函数，一个处理服务器响应的函数，使用 onreadystatechange，类似函数指针

（3）获取异步对象的readyState 属性：该属性存有服务器响应的状态信息。每当 readyState 改变时，onreadystatechange 函数就会被执行。

（4）判断响应报文的状态，若为200说明服务器正常运行并返回响应数据。

（5）读取响应数据，可以通过 responseText 属性来取回由服务器返回的数据。

### 如何实现ajax请求，假如我有多个请求，我需要让这些ajax请求按照某种顺序一次执行，有什么办法呢？如何处理ajax跨域

通过实例化一个XMLHttpRequest对象得到一个实例，调用实例的open() 为这次ajax请求设定相应的http方法get，相应的地址url和是否异步true或false。以异步为例，调用send()方法，这个方法可以设定需要发送的报文主体。然后通过监听readystatechange（准备状态改变）事件，通过这个实例的readyState（准备状态）属性来判断这个ajax请求状态，其中分为0，1，2，3，4这四种状态（0未初始化，1载入/正在发送请求2载入完成/数据接收，3交互/解析数据，4接收数据完成），当状态为4的时候也就是接受数据完成的时候，这时候可以通过实例的status属性判断这个请求是否成功。

var xhr = new XMLHttpRequest();//创建xhr对象

xhr.open('get', 'aabb.php', true);//http方法为get，相应地址url是这个，是异步的

xhr.send(null);//发送报文主体

xhr.onreadystatechange = function() {//监听准备状态事件

if(xhr.readyState==4) {//如果是4，也就是说解析完成，可以在客户端调用了

if(xhr.status==200) {

//如果状态码是200，服务器正常运行，并返回相应数据（请求成功，get和post方法）

console.log(xhr.responseText);//打印出相应的数据

}}}

**使ajax请求按照队列顺序执行，通过调用递归函数：**

//按顺序执行多个ajax命令，因为数量不定，所以采用递归

**function send(action, arg2) {**

//将多个命令按顺序封装成数组对象，递归执行

//利用了deferred对象控制回调函数的特点

**$.when(send\_action(action[0], arg2))**

**.done(function () {**

//前一个ajax回调函数完毕之后判断队列长度

**if (action.length > 1) {**

//队列长度大于1，则弹出第一个，继续递归执行该队列

**action.shift();**

**send(action, arg2);**

**}**

**}).fail(function (){**

//队列中元素请求失败后的逻辑

//重试发送

//send(action, arg2);

//忽略错误进行下个

//if (action.length > 1) {

//队列长度大于1，则弹出第一个，继续递归执行该队列

//    action.shift();

//    send(action, arg2);

//}

**});}**

//处理每个命令的ajax请求以及回调函数

**function send\_action(command, arg2) {**

**var dtd = $.Deferred();//定义deferred对象**

**$.post(**

**"url",**

**{command: command,**

**arg2: arg2}**

**).done(function (json) {**

**json = $.parseJSON(json);**

//每次请求回调函数的处理逻辑

//逻辑结束

**dtd.resolve();**

**}).fail(function (){**

//ajax请求失败的逻辑

**dtd.reject();**

**});**

**return dtd.promise();**

//返回Deferred对象的promise，防止在外部修改状态

**}**

### 写出原生Ajax

Ajax能够在**不重新加载整个页面**的情况下与服务器交换数据并**更新部分网页内容**，实现**局部刷新**，大大**降低了资源的浪费**，是一门用于**快速创建动态网页的技术**，ajax的使用分为四部分：

1、创建XMLHttpRequest对象var xhr = new XMLHttpRequest();

2、向服务器发送请求，使用xmlHttpRequest对象的open和send方法，

3、监听状态变化onreadystatechange，执行相应回调函数readyState&&status

var xhr = new XMLHttpRequest();

xhr.open('get', 'aabb.php', true);

xhr.send(null);

xhr.onreadystatechange = function() {

if(xhr.readyState==4) {

if(xhr.status==200) {

console.log(xhr.responseText);}}}

### 如何实现一个ajax请求？如果我想发出两个有顺序的ajax需要怎么做？

AJAX创建异步对象XMLHttpRequest

操作XMLHttpRequest对象

（1）设置请求参数open（请求方式get post，请求页面的相对路径url，是否异步true false）、send（数据包）

（2）设置回调函数onreadystatechange，用于处理服务器响应的函数，并监听 onreadystatechange，类似函数指针

（3）获取异步对象的readyState属性：该属性存有服务器响应的状态信息。每当 readyState 改变时，onreadystatechange 函数就会被执行。

（4）判断响应报文的状态，若readyState为4且status为200说明服务器正常运行并返回响应数据。

（5）读取响应数据，可以通过 responseText 属性来取回由服务器返回的数据。

发出两个有顺序的ajax，可以用**回调函数**，也可以使用**Promise.then**或者**async**等。

### Fetch和Ajax比有什么优缺点？

fetch()是 XMLHttpRequest 的升级版，用于在 JavaScript 脚本里面发出 HTTP 请求。fetch()使用 Promise，不使用回调函数。promise方便异步，在不想用jQuery的情况下，相比原生的ajax，也比较好写。

### 原生JS的ajax

AJAX创建异步对象XMLHttpRequest

操作XMLHttpRequest 对象

（1）设置请求参数open（请求方式get或post，请求页面的相对路径url，是否异步true或false）

（2）设置回调函数onreadystatechange，并监听这个函数，类似函数指针

（3）获取异步对象的readyState 属性：该属性存有服务器响应的状态信息。每当 readyState 改变时，onreadystatechange 函数就会被执行。

（4）判断响应报文的状态，若为200说明服务器正常运行并返回响应数据。

（5）读取响应数据，可以通过 responseText 属性来取回由服务器返回的数据。

### 知道PWA吗

PWA全称Progressive Web App，即渐进式WEB应用。一个 PWA 应用首先是一个网页, 可以通过 Web 技术编写出一个网页应用. 随后添加上 App Manifest 和 Service Worker 来实现 PWA 的安装和离线等功能。 Web App 和 Native App 各自的优势融合在一起，让用户在使用 Web App 时感觉在使用 Native App