什么是跨域？

概念：只要协议、域名、端口有任何一个不同，都被当作是不同的域。

URL 说明 是否允许通信

http://www.a.com/a.js

http://www.a.com/b.js 同一域名下 允许

http://www.a.com/lab/a.js

http://www.a.com/script/b.js 同一域名下不同文件夹 允许

http://www.a.com:8000/a.js

http://www.a.com/b.js 同一域名，不同端口 不允许

http://www.a.com/a.js

https://www.a.com/b.js 同一域名，不同协议 不允许

http://www.a.com/a.js

http://70.32.92.74/b.js 域名和域名对应ip 不允许

http://www.a.com/a.js

http://script.a.com/b.js 主域相同，子域不同 不允许

http://www.a.com/a.js

http://a.com/b.js 同一域名，不同二级域名（同上） 不允许（cookie这种情况下也不允许访问）

http://www.cnblogs.com/a.js

http://www.a.com/b.js 不同域名 不允许

对于端口和协议的不同，只能通过后台来解决。

跨域资源共享（CORS）

CORS（Cross-Origin Resource Sharing）跨域资源共享，定义了必须在访问跨域资源时，浏览器与服务器应该如何沟通。CORS背后的基本思想就是使用自定义的HTTP头部让浏览器与服务器进行沟通，从而决定请求或响应是应该成功还是失败。

<script type="text/javascript">

**var** xhr = **new** XMLHttpRequest();

xhr.open("￼GET", "/trigkit4",true);

xhr.send();

</script>

以上的trigkit4是相对路径，如果我们要使用CORS，相关Ajax代码可能如下所示：

<script type="text/javascript">

**var** xhr = **new** XMLHttpRequest();

xhr.open("￼GET", "http://segmentfault.com/u/trigkit4/",true);

xhr.send();

</script>

代码与之前的区别就在于相对路径换成了其他域的绝对路径，也就是你要跨域访问的接口地址。

服务器端对于CORS的支持，主要就是通过设置Access-Control-Allow-Origin来进行的。如果浏览器检测到相应的设置，就可以允许Ajax进行跨域的访问。

要解决跨域的问题，我们可以使用以下几种方法：

通过jsonp跨域

现在问题来了？什么是jsonp？维基百科的定义是：JSONP（JSON with Padding）是资料格式 JSON 的一种“使用模式”，可以让网页从别的网域要资料。

JSONP也叫填充式JSON，是应用JSON的一种新方法，只不过是被包含在函数调用中的JSON，例如：

callback({"name","trigkit4"});

JSONP由两部分组成：回调函数和数据。回调函数是当响应到来时应该在页面中调用的函数，而数据就是传入回调函数中的JSON数据。

在js中，我们直接用XMLHttpRequest请求不同域上的数据时，是不可以的。但是，在页面上引入不同域上的js脚本文件却是可以的，jsonp正是利用这个特性来实现的。 例如：

<script type="text/javascript">

**function** dosomething(jsondata){

//处理获得的json数据

}

</script>

<script src="http://example.com/data.php?callback=dosomething"></script>

js文件载入成功后会执行我们在url参数中指定的函数，并且会把我们需要的json数据作为参数传入。所以jsonp是需要服务器端的页面进行相应的配合的。

**<?php**

$callback = $\_GET['callback'];//得到回调函数名

$data = **array**('a','b','c');//要返回的数据

**echo** $callback.'('.json\_encode($data).')';//输出

**?>**

最终，输出结果为：dosomething(['a','b','c']);

如果你的页面使用jquery，那么通过它封装的方法就能很方便的来进行jsonp操作了。

<script type="text/javascript">

$.getJSON('http://example.com/data.php?callback=?,function(jsondata)'){

//处理获得的json数据

});

</script>

jquery会自动生成一个全局函数来替换callback=?中的问号，之后获取到数据后又会自动销毁，实际上就是起一个临时代理函数的作用。$.getJSON方法会自动判断是否跨域，不跨域的话，就调用普通的ajax方法；跨域的话，则会以异步加载js文件的形式来调用jsonp的回调函数。

JSONP的优缺点

JSONP的优点是：它不像XMLHttpRequest对象实现的Ajax请求那样受到同源策略的限制；它的兼容性更好，在更加古老的浏览器中都可以运行，不需要XMLHttpRequest或ActiveX的支持；并且在请求完毕后可以通过调用callback的方式回传结果。

JSONP的缺点则是：它只支持GET请求而不支持POST等其它类型的HTTP请求；它只支持跨域HTTP请求这种情况，不能解决不同域的两个页面之间如何进行JavaScript调用的问题。

CORS和JSONP对比

CORS与JSONP相比，无疑更为先进、方便和可靠。

1、 JSONP只能实现GET请求，而CORS支持所有类型的HTTP请求。

2、 使用CORS，开发者可以使用普通的XMLHttpRequest发起请求和获得数据，比起JSONP有更好的错误处理。

3、 JSONP主要被老的浏览器支持，它们往往不支持CORS，而绝大多数现代浏览器都已经支持了CORS）。

通过修改document.domain来跨子域

浏览器都有一个同源策略，其限制之一就是第一种方法中我们说的不能通过ajax的方法去请求不同源中的文档。 它的第二个限制是浏览器中不同域的框架之间是不能进行js的交互操作的。  
不同的框架之间是可以获取window对象的，但却无法获取相应的属性和方法。比如，有一个页面，它的地址是http://www.example.com/a.html ， 在这个页面里面有一个iframe，它的src是http://example.com/b.html, 很显然，这个页面与它里面的iframe框架是不同域的，所以我们是无法通过在页面中书写js代码来获取iframe中的东西的：

<script type="text/javascript">

**function** test(){

**var** iframe = document.getElementById('￼ifame');

**var** win = document.contentWindow;//可以获取到iframe里的window对象，但该window对象的属性和方法几乎是不可用的

**var** doc = win.document;//这里获取不到iframe里的document对象

**var** name = win.name;//这里同样获取不到window对象的name属性

}

</script>

<iframe id = "iframe" src="http://example.com/b.html" onload = "test()"></iframe>

这个时候，document.domain就可以派上用场了，我们只要把http://www.example.com/a.html 和http://example.com/b.html这两个页面的document.domain都设成相同的域名就可以了。但要注意的是，document.domain的设置是有限制的，我们只能把document.domain设置成自身或更高一级的父域，且主域必须相同。

1.在页面 http://www.example.com/a.html 中设置document.domain:

<iframe id = "iframe" src="http://example.com/b.html" onload = "test()"></iframe>

<script type="text/javascript">

document.domain = 'example.com';//设置成主域

**function** test(){

alert(document.getElementById('￼iframe').contentWindow);//contentWindow 可取得子窗口的 window 对象

}

</script>

2.在页面 http://example.com/b.html 中也设置document.domain:

<script type="text/javascript">

document.domain = 'example.com';//在iframe载入这个页面也设置document.domain，使之与主页面的document.domain相同

</script>

修改document.domain的方法只适用于不同子域的框架间的交互。

使用window.name来进行跨域

window对象有个name属性，该属性有个特征：即在一个窗口(window)的生命周期内,窗口载入的所有的页面都是共享一个window.name的，每个页面对window.name都有读写的权限，window.name是持久存在一个窗口载入过的所有页面中的

使用HTML5的window.postMessage方法跨域

window.postMessage(message,targetOrigin) 方法是html5新引进的特性，可以使用它来向其它的window对象发送消息，无论这个window对象是属于同源或不同源，目前IE8+、FireFox、Chrome、Opera等浏览器都已经支持window.postMessage方法。

本人兼职前端付费技术顾问，如需帮助请加本人微信hawx1993，非诚勿扰

1.为初学前端而不知道怎么做项目的你指导

2.指导并扎实你的JavaScript基础

3.帮你准备面试并提供相关指导性意见

4.为你的前端之路提供极具建设性的意见

5.让你的学习更有效率，对知识点的理解更加透彻

6.不解答可以百度到的答案，也不解答你遇到的bug

付费（每月仅需99）能让我更有耐心和责任心的为你指导，让双方产生更良好的协作