跨域：如果是跨源（非同源地址）的请求，受同源策略的影响，就无法得到响应结果。跨域并不是请求发不出去，请求能发出去，服务端能收到请求并正常返回结果，只是结果被浏览器拦截了。

同源策略（SOP）：同源策略是一种约定，它是浏览器最核心也最基本的安全功能，如果缺少了同源策略，浏览器很容易受到 XSS、CSFR 等攻击，同源，就是协议名、主机名、端口号完全相同；

http:// www . abc.com : 8080 / jquery.js

协议 子域名 主域名 端口号 请求资源地址

ajax跨域表现：No 'Access-Control-Allow-Origin' header is present on the requested resource

解决ajax跨域问题：

1. 通过jsonp跨域：
   1. 原理：JSONP的核心则是动态添加<script>标签来调用服务器提供的js文件。
      1. 同源策略仅仅是对JS代码进行限制，带有src属性的HTML标签属性值是一个服务器地址，这些标签会从该地址取得服务器的数据。更重要的是，这些HTML标签对服务器的请求，不会受到同源策略的限制！
      2. 可以在远程服务器上设法把数据装进js格式的文件里，供客户端调用处理，实现跨域。
      3. JSONP协议允许用户传递一个callback参数给服务端，然后服务端返回数据时会将这个callback参数作为函数名来包裹住JSON数据，这样客户端就可以随意定制自己的函数来自动处理返回数据了。
   2. 具体实现：



* 1. 缺点：只能实现get一种请求，不安全可能会遭受 XSS 攻击；

1. document.domain + iframe跨域：
   1. 原理：两个页面都通过js强制设置document.domain为基础主域，就实现了同域；
   2. 实现：



* 1. 缺点：仅限主域相同，子域不同的跨域应用场景；

1. location.hash + iframe
2. window.name + iframe跨域
3. postMessage跨域：HTML5 XMLHttpRequest Level 2 中的 API，且是为数不多可以跨域操作的 window 属性之一；
4. 跨域资源共享（CORS）：为服务器的响应头中，新增了一个重要的键，它是Access-Control-Allow-Origin；在域名位置写上星号，表示服务器的响应结果可以被任意源访问；
   1. 实现原理：它允许浏览器向跨源[服务器](http://link.zhihu.com/?target=https://www.baidu.com/s?wd=%E6%9C%8D%E5%8A%A1%E5%99%A8&tn=24004469_oem_dg&rsv_dl=gh_pl_sl_csd" \t "https://zhuanlan.zhihu.com/p/_blank)，发出XMLHttpRequest请求，从而克服了AJAX只能同源使用的限制，后端是实现 CORS 通信的关键；
   2. CORS请求：
      1. 简单请求：同时满足一下条件：
         1. 请求方法：HEAD、POST、GET;
         2. HTTP的头信息不超出以下几种字段：Accept、Accept-Language、Content-Language、Last-Event-ID、Content-Type；
      2. 非简单请求：其他；
5. nginx代理跨域
6. nodejs中间件代理跨域
7. WebSocket协议跨域：Websocket 是 HTML5 的一个持久化的协议，它实现了浏览器与服务器的全双工通信，同时也是跨域的一种解决方案。WebSocket 和 HTTP 都是应用层协议，都基于 TCP 协议。但是WebSocket 是一种双向通信协议，在建立连接之后，WebSocket 的 server 与 client 都能主动向对方发送或接收数据；