1. 前端跨域

$.ajax({

url: 'http://www.domain2.com:8080/login',

type: 'get',

dataType: 'jsonp', // 请求方式为jsonp

jsonpCallback: "handleCallback", // 自定义回调函数名

data: {}});

1. 通过jsonp跨域,缺点：只能实现get一种请求。
2. **document.domain** + iframe跨域:此方案仅限主域相同，子域不同的跨域应用场景。

实现原理：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。

1.）父窗口：(http://www.domain.com/a.html)

<iframe id="iframe" src="http://child.domain.com/b.html"></iframe>

<script> document.domain = 'domain.com'; var user = 'admin' </script>

1. ）子窗口：(http://child.domain.com/b.html)

<script> document.domain = 'domain.com'; // 获取父窗口中变量

alert('get js data from parent ---> ' + window.parent.user); </script>

1. **location.hash** + iframe跨域:原理：a与b跨域只能通过hash值单向通信，通过中间页c来实现。 三个页面，不同域之间利用iframe的location.hash传值，相同域(a.c)之间直接js访问来通信。
2. **window.name** + iframe跨域
3. **跨域资源共享（CORS）**
4. postMessage跨域 7)nginx代理跨域

8)nodejs中间件代理跨域 9)WebSocket协议跨域

2.<iframe> 标签是一个**内联框架**，即用来在当前 HTML 页面中嵌入另一个文档的，且所有主流浏览器都支持iframe标签。语法：<iframe src="文件路径"></iframe>

src用于设置框架的地址，可以使页面地址，也可以是图片地址

1. Vue是一套用于构建用户界面的渐进式框架=>在开发过程中来按需引入功能，不是一次性都放进来的，有点类似于迭代开发
2. v-if v-for区别