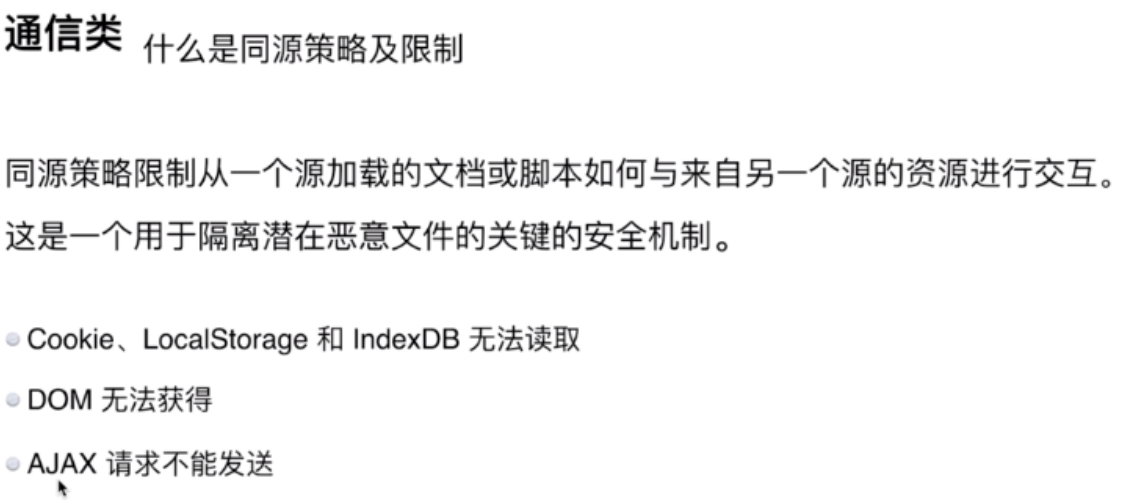
跨域通信常用方式：J
SONP，
Hash(hash的改变不会刷新页面)
postMessage，（h5）
WebSocket，
CORS

前后端如何通信：
ajax（同源）
websocket（不受同源限制）
CORS（同源、跨域都支持）

同源策略
源：包含协议、域名、端口；三者有一个不一样就跨域了




```
// 利用hash，场景是当前页面 A 通过iframe或frame嵌入了跨域的页面 B
    // 在A中伪代码如下：
    var B = document.getElementsByTagName('iframe');
    B.src = B.src + '#' + 'data';
    // 在B中的伪代码如下
    window.onhashchange = function () {
        var data = window.location.hash;
    };

    // postMessage
    // 窗口A(http:A.com)向跨域的窗口B(http:B.com)发送信息
    Bwindow.postMessage('data', 'http://B.com');
    // 在窗口B中监听
    Awindow.addEventListener('message', function (event) {
        console.log(event.origin);
        console.log(event.source);
        console.log(event.data);
    }, false);

    // Websocket【参考资料】
http://www.ruanyifeng.com/blog/2017/05/websocket.html
```

```
var ws = new WebSocket('wss://echo.websocket.org');

ws.onopen = function (evt) {
    console.log('Connection open ...');
    ws.send('Hello WebSockets!');
};

ws.onmessage = function (evt) {
    console.log('Received Message: ', evt.data);
    ws.close();
};

ws.onclose = function (evt) {
    console.log('Connection closed.');
};

// CORS【参考资料】http://www.ruanyifeng.com/blog/2016/04/cors.html
// url（必选），options（可选）
fetch('/some/url/', {
    method: 'get',
}).then(function (response) {

}).catch(function (err) {
  // 出错了，等价于 then 的第二个参数，但这样更好用更直观
});
```