

# 面试准备

## 1. 什么是跨域，如何解决跨域？

浏览器不能执行其他网站的脚本，由浏览器同源策略引起的，是对javascript施加的安全限制。同源指的是域名、端口、协议相同。解决方法：1.设置代理 2.修改请求头设置来源访问为所有，访问方式为相应的方式。

## 2. javascript放在head和body里的不同？

javascript放在head里需要等所有js代码下载完成后才能进行页面的渲染，现代的web应用程序都会把js放在body的最后，这样能够先渲染页面在下载js代码，对用户来讲，感觉上速度加快了。

## 3. js使用外部引用的好处？

1. 在支持SPDY/HTTP2的浏览器中，当多个文件引入同一个文件的时候，采用外部引用浏览器会缓存js文件，导致在其他页面中，引入的同一个文件，一定程度上加快运行速度
2. 外部引用可以更方便的维护

## 4. 浏览器缓存机制

缓存的位置：

### Service Worker

首先注册service worker，监听到install事件后就可以缓存需要的文件，下次用户访问的时候，采用拦截请求的方式在缓存查询，有的话返回，没有的话，去请求数据

### Memory Cache

### Disk Cache

### Push Cache

只在会话（Session）中存在，一旦会话结束就被释放，并且缓存时间也很短暂

浏览器的缓存机制：第一次http请求发送给浏览器cache，查询是否有缓存结果和缓存标识，如果没有向服务器发送第二次http请求，返回结果和缓存规则，并将其添加到浏览器

的缓存

浏览器的缓存策略：强缓存和协商缓存，都是通过http Header实现的。

## 强缓存

从服务器的缓存中读取资源。Size显示from disk cache或者 from memory cache。实现方法：Expires、Cache-Control

Expires:(http1) 受限于本地时间。

Cache-Control(http1.1):

public: 可以被客户端、代理服务器缓存

private: 响应只可以被客户端缓存

max-age: 缓存30秒过期，需要重新请求

no-store: 不缓存

no-cache: 客户端缓存内容，是否使用则需要经过协商缓存来验证决定。使用缓存数据时，需要确认数据是否和服务器保持一致。

## 协商缓存

强制缓存失效后，浏览器携带缓存标识向服务器发送请求，由服务器根据缓存标识决定是否使用缓存的过程。

304 Not Modified

200 Modified

实现方法：Last-Modified 和 ETag

Last-Modified 和If-Modified-Since联合使用。

Last-Modified弊端：1. 如果本地打开缓存文件，也会修改Last-Modified的值，导致浏览器发送相同的资源。2. Last-Modified用s来计时，在不可感知的事件内修改文件，服务器会认为资源没有被修改，不会返回正确资源

ETag和IF-None-Match

服务器为每个资源生成唯一的ETag，浏览器请求的时候，在IF-None-Match中携带ETAG的值，如果不同，返回200，否则返回304

如果什么缓存策略都没设置，那么浏览器会怎么处理？

浏览器会采用一个启发式的算法，通常会取响应头中的 Date 减去 Last-Modified 值的 10% 作为缓存时间。

## 5. 什么时XSS，防止XSS攻击

防止XSS攻击：

1. 设置HttpOnly, 这样只可以通过request.getCookies()获得cookie

## 6. Vue实现双向绑定的原理

1. 创建虚拟dom树document.createDocumentFragment()创建虚拟树
2. 一旦检测到数据变化，通过Object.defineProperty定义分数据拦截，获取到数据变化。
3. 通过订阅-发布者模式，触发观察者，改变虚拟dom树中的具体数据。
4. 最后渲染到dom树中

## 7. 什么是http？

http协议是基于TCP-IP的报文协议，报文头是不定长任意扩展的。

## 8. meta中的viewport有哪些参数？

width/Height 设置长宽

initial-scale: 缩放比

user-scalable: 用户是否可放大

## 9. vue2.x中如何监测数组变化

采用函数劫持的方法，重新定义了数组的方法，对data中的数组进行了原型链重写，指向了自己定义的数组原型，这样当调用数组的api的时候，可以通知依赖更新，当数组中有引用类型的时候，会对引用类型进行递归遍历监控，从而实现数组的监控

## 10. nextTick知道吗，实现原理是什么？

nextTick 是异步更新策略，在下次dom更新循环结束之后，进行延迟回调。目前浏览器没有支持nextTick，vue使用promise、setTimeout、setImmediate来实现，多次调用nextTick，就将回调函数存在队列中，只有当前栈的任务都执行完，采取执行队列中的函数

## 11. 组件中的data为什么是一个函数

一个组件被复用多次的话，也就会创建多个实例。本质上，这些实例用的都是同一个构造函数。如果data是对象的话，对象属于引用类型，会影响到所有的实例。所以为了保证组件不同的实例之间data不冲突，data必须是一个函数

## 12. vue绑定函数原理

1. 原生的dom和组件上有.native修饰符最终调用的是原生js的addEventListener
2. Vue实例上的事件会调用原型的\$on, \$emit等方法