一面

1.XSS CSRF 如何防止

2.页面性能优化 首屏加载的时间如何统计

3.node拍页面的方式，有什么优势

二面

1.TCP三次握手 丢包重传 客户端如何判断包重复 滑动窗口的大小如何调整 UDP不用建立连接 UDP和TCP的效率问题 ，HTTP，HTTPS

1） Client首先发送一个连接试探，ACK=0 表示确认号无效，SYN = 1 表示这是一个连接请求或连接接受报文，同时表示这个数据报不能携带数据，seq = x 表示Client自己的初始序号（seq = 0 就代表这是第0号包），这时候Client进入syn_sent状态，表示客户端等待服务器的回复2） Server监听到连接请求报文后，如同意建立连接，则向Client发送确认。TCP报文首部中的SYN 和 ACK都置1 ，ack = x 1表示期望收到对方下一个报文段的第一个数据字节序号是x 1，同时表明x为止的所有数据都已正确收到（ack=1其实是ack=0 1,也就是期望客户端的第1个包），seq = y 表示Server 自己的初始序号（seq=0就代表这是服务器这边发出的第0号包）。这时服务器进入syn_rcvd，表示服务器已经收到Client的连接请求，等待client的确认。3） Client收到确认后还需再次发送确认，同时携带要发送给Server的数据。ACK 置1 表示确认号ack= y 1 有效（代表期望收到服务器的第1个包），Client自己的序号seq= x 1（表示这就是我的第1个包，相对于第0个包来说的），一旦收到Client的确认之后，这个TCP连接就进入Established状态，就可以发起http请求了。

每次连接后seq都+1，可以根据这个判断是否有丢包或者重复

关于TCP的重传机制：http://www.cnblogs.com/jackdong/archive/2010/08/09/1796092.html

与数据链路层的ARQ协议相类似，TCP使用超时重发的重传机制。即：TCP每发送一个报文段，就对此报文段设置一个超时重传计时器。此计时器设置的超时重传时间**RTO**（Retransmission Time－Out）应当略大于TCP报文段的平均往返时延RTT，一般可取**RTO**＝2RTT。但是，也可以根据具体情况人为**调整RTO**的值，例如可以设置此超时重传时间**RTO**＝90秒。当超过了规定的超时重传时间还未收到对此TCP报文段的预期确认信息，则必须重新传输此TCP报文段。注意，TCP在使用滑动窗口时，可以等效为数据链路层讨论过的连续ARQ的情况。因此某TCP报文段超时，则只重传此报文段。而其后已经成功传送的报文段不在此重传的范围。

HTTPS：https://blog.wilddog.com/?p=210

2. 数据结构，快速排序 时间复杂度nlogn的推导

3. 作用域链与原型链

4. 浏览器缓存的机制，各种方式的优缺点，比如Lat-Modified适合于一些静态资源

http://www.cnblogs.com/skynet/archive/2012/11/28/2792503.html

三面
1.JSONP超时，Comet实现，Websocket协议
如何确定JSONP请求超时：

如果依赖于zepto，设置一个定时器，当jsonp超时的时候，可以通过abort方法：

Websocket不采用同源策略，但可以通过Websocket请求头的Origin来管理：

2.MVVM，比如avalon是怎么做双向绑定的
http://blog.gejiawen.com/2015/04/02/2-way-data-binding-and-define-property/

3.哪些浏览器兼容性的问题

4.详细讲解一个项目中遇到的问题
汽车票列表页：
1.模块间通信 发布订阅
2.列表的分页加载，loading，判断元素是否在视窗范围内
3.使用百度地图api，节点缓存，对象池；取不到offsetParent,safari全屏需要计算innerHeight差值
4.无痕模式下的错误，localstorage
5.学会画类图，拆分成原子操作
6.for..in..排序的问题
7.IOS下click的冒泡

```
var $attr = function(id, name, value) {
    var dom = document.getElementById(id),
        arr = [];
    if (!dom) {
        return arr;
    }
    getAttr(dom,name,value);
    return arr;

    function getAttr(dom, name, value) {
        if (dom.hasAttribute(name) && dom.getAttribute(name) == value) {
            arr.push(dom);
        }
        if (dom.hasChildNodes()) {
            var childnodes = dom.childNodes;
            for (var i = 0; i < childnodes.length; i++) {
                if (childnodes[i].nodeType == 1) {
                    getAttr(childnodes[i],name,value);
                }
            }
        }
    }

};
```

四面+笔试

1.ul和一堆li，click输出li的内容、坐标；鼠标离开ul的时候，弹窗，注意兼容

事件冒泡（这里需要注意如果li还有子元素比如p，那点击p元素的时候冒泡到ul上的e.target是p，需要通过parentNode往上找），离开ul的事件绑定mouseleave，mouseleave是DOM3的标准，不冒泡且不再后代元素上触发

2. 实现一个$attr(domId,name,value)遍历id是domId的元素内部属性为name且值为value的元素

3. 用json和xml描述一个对象

4. 写一个原生的xhr请求过程，readystate的几种状态分别什么含义，以及怎么跨域

5. 两种以上实现数组去重

对象hash、ES5的fliter和ES6的Set

6.写快速排序

快排 http://www.ruanyifeng.com/blog/2011/04/quicksort_in_javascript.html

```
var array = [5, 4, 13, 6, 9, 17, 19, 22, 31, 1, 99, 145];
        function quickSort(arr) {
            if (arr.length <= 1) return arr;
            var s = arr.shift();
            var left = [];
            var right = [];
            for (var i = 0; i < arr.length; i++) {
                if (arr[i] < s) left.push(arr[i]);
                else right.push(arr[i]);
            }
            return quickSort(left).concat([s],quickSort(right));
        }
        var a=quickSort(array);
        alert(a);
```

7. 遍历一个大的table，逆序输出table里面单元格的内容，table可能嵌套

使用js的treewalk