# 腾讯面试（一面、二面）

时间：2019 年 3 月 22 日 部门：腾讯新闻

## 手撕代码

### 1.写 call 函数的实现

Function.prototype.call2 = function (context) {  
 var context = context || window;  
 context.fn = this;  
 var args = [];  
 for(var i = 1, len = arguments.length; i < len; i++) {  
 args.push('arguments[' + i + ']');  
 }  
 var result = eval('context.fn(' + args +')');  
 delete context.fn  
 return result;  
}

### 2.js 实现数组千分位

//正则实现  
function format (num) {  
 var reg=/\d{1,3}(?=(\d{3})+$)/g;  
 return num.toString().replace(reg, '$&,');  
}  
//基础  
function format(num){  
 num+='';  
 var str="";  
 for(var i=num.length-1,j=1;i>=0;i--,j++){  
 if(j%3===0 & i!=0){  
 str+=num[i]+',';  
 }else{  
 str+=num[i];  
 }  
 }  
 return str.split('').reverse().join('');  
}

### 3. 手写快速排序算法

var quickSort = function(arr) {  
 if(arr.length<=1)  
 return arr;  
 var left=[],right=[];  
 var index=Math.floor(arr.length/2);  
 var midVal=arr.splice(midVal,1)[0];  
 for(var i=0;i<arr.length;i++){  
 arr[i]<midVal?left.push(arr[i]):right.push(arr[i]);  
 }  
 return quickSort(left).concat(midVal).concat(quickSort(right));  
};

### 4. 查找数组中元素和等于给定数的子数组

var ans,res,len;  
var dfs=function(index,sum,candidates,target){  
 if(sum===target){  
 var tmp=res.map(function(item){  
 return item;  
 })  
 ans.push(tmp);  
 // console.log(res,ans);  
 return ;  
 }  
 for(var i=index;i<len;i++){  
 if(sum+candidates[i]>target)  
 continue;  
 res.push(candidates[i]);  
 dfs(i,sum+candidates[i],candidates,target);  
 res.pop();  
 }  
}  
var combinationSum = function(candidates, target) {  
 ans=[];  
 len=candidates.length;  
 candidates.sort((a,b)=>a-b);  
 for(var i=0;i<len;i++){  
 res=[candidates[i]];  
 dfs(i,candidates[i],candidates,target);  
 }  
 return ans;  
};

## vue

### 1. vuex 原理

* vuex 的 store 有 State、 Getter、Mutation 、Action、 Module 五种属性；
* **state** 为单一状态树，在 state 中需要定义我们所需要管理的数组、对象、字符串等等
* **getters** 类似 vue 的计算属性，主要用来过滤一些数据。
* **mutation** 更改 store 中 state 状态的唯一方法就是提交 mutation，store.commit。
* **action** actions 可以理解为通过将 mutations 里面处里数据的方法变成可异步的处理数据的方法，简单的说就是异步操作数据。view 层通过 store.dispath 来分发 action。
* **module** module 其实只是解决了当 state 中很复杂臃肿的时候，module 可以将 store 分割成模块，每个模块中拥有自己的 state、mutation、action 和 getter。

### 2. vue 数据双向绑定

<body>  
 <div id="app">  
 <input type="text" id="txt">  
 <p id="show"></p>  
</div>  
</body>  
<script type="text/javascript">  
 var obj = {}  
 Object.defineProperty(obj, 'txt', {  
 get: function () {  
 return obj  
 },  
 set: function (newValue) {  
 document.getElementById('txt').value = newValue  
 document.getElementById('show').innerHTML = newValue  
 }  
 })  
 document.addEventListener('keyup', function (e) {  
 obj.txt = e.target.value  
 })  
</script>

### 3. 父子组件如何通信，兄弟组件如何通信

### 4.vue 如何将一个组件打包上传

## 安全

### 1. 如何解决浏览器劫持问题

### 2. 讲讲 CSRF 和 XSS 区别

### 3. DNS 劫持与 DNS 污染

* **DNS 劫持** DNS 劫持就是通过劫持了 DNS 服务器，通过某些手段取得某域名的解析记录控制权，进而修改此域名的解析结果，导致对该域名的访问由原 IP 地址转入到修改后的指定 IP。DNS 劫持通过篡改 DNS 服务器上的数据返回给用户一个错误的查询结果来实现的。
* **DNS 污染** DNS 污染，指的是用户访问一个地址，国内的服务器(非 DNS)监控到用户访问的已经被标记地址时，服务器伪装成 DNS 服务器向用户发回错误的地址的行为。范例，访问 Youtube、Facebook 之类网站等出现的状况。

## 网络

### 1.在浏览器输入网址，知道页面出现发生了啥？

## 性能优化

### 1.计算浏览器的白屏时间

### 2. 加载一个很长的列表，怎么优化性能？

## JavaScript

### 讲讲对 promise 的理解？

# 腾讯面试（一面、二面、三面）

时间：2019 年 4 月 1 日 部门：腾讯地图

### 写个继承

### 记忆最深的项目

### 未来职业规划

### 留不留北京，要不要户口

### 如何看待加班

### vue 双向数据绑定，讲讲观察者订阅者模式

### 讲讲对 d3 的理解，讲讲 d3 与 echarts 的区别

* d3 正如其名 Data Driven Documents，其本质是将数据与 DOM 绑定，并将数据映射至 DOM 属性上；
* d3 与 echarts 的区别：

1. d3 通过 svg 绘制图形，可以自定义事件。svg 不依赖分辨率，继续 xml 绘制图形，可以操作 dom。支持事件处理器，复杂度高，会减慢页面的渲染速度。
2. echarts 通过 canvas 来绘制图形，用户通过配置 options 参数，就可很容易绘制指定图表。canvas 依赖分辨率，基于 js 绘制图形，不支持事件处理，能以 png 或者 jpg 的格式保存图片。

# 腾讯 hr 面

时间：2019 年 4 月 18 日 方式：电话

### 自我介绍

### 印象最深刻的一个项目，为什么，这个项目给你的最大的收获是啥？

### 近期遇到的最大的困难是啥？如何解决？期间是如何努力的？

### 最近印象最深刻的事情是啥？

### 学习能力强的原因是啥？

### 有什么业余爱好？

### 最近去哪些地方玩过？