1. Js的兼容性问题
2. 冒泡
3. 页面优化
4. 网络分层

应用层、表示层、会话层、传输层、网络层、数据链路层、物理层

1. 三栏布局
2. 浮动后果、清除浮动
3. Flex
4. 链表和队列的区别
5. 反向二叉树
6. 网格布局
7. 死锁
8. Js继承
9. For循环Settimeout
10. jQuery链式调用
11. 一个空的div实现一半一半
12. 动态网页
13. MVVM
14. TCP
15. TCP（Transmission Control Protocol 传输控制协议）是一种面向连接的、可靠的、基于字节流的传输层通信协议

### http/html浏览器

1. 说一下http和https
2. Tcp三次握手，一句话描述
3. TCP和UDP区别
4. Websocket的实现和应用
5. http请求的方式，head方式
6. 一个图片url访问后直接下载这样实现
7. 说一下web quality（无障碍）
8. 几个实用的bom属性对象方法
9. 说一下HTML5 drag api
10. 说一下http2.0，http2.0的特性
11. 补充400和401、403状态码
12. Fetch发送2次请求的原因
13. Cookie、sessionstorage、localstorage的区别
14. 说一下web worker
15. 对HTML语义化标签的理解
16. Iframe是什么？有什么缺点
17. Doctype作用？严格模式与混杂模式如何区分？它们有何意义？
18. Cookie如何防范xss攻击
19. Cookie和session的区别，localstorage和sessionstorage的区别
20. 一句概括RESTFUL
21. 讲讲viewport和移动端布局
22. Click在iOS上有300ms延迟，原因及如何解决？
23. Addeventlistener参数
24. Iframe通信，同源和不同源两种情况，多少种方法。
25. 介绍知道http返回的状态码
26. http常用请求头
27. 强、协商缓存
28. 讲讲304
29. 强缓存、协商缓存什么时候用哪个
30. 前端优化
31. GET和POST的区别
32. 301和302的区别，说说302、301、304状态码
33. http支持的方法
34. 如何画一个三角形
35. 状态码304和200，既然你看过图解http，那你回答下200和304的区别
36. 说一下浏览器缓存
37. HTML5新增的元素
38. 在地址栏里输入一个URL，到这个页面呈现出来，中间会发生什么？
39. 常见的http的头部
40. Cache-control的值有哪些
41. 浏览器在生成页面的时候，会生成哪两棵树
42. Csrf和xss的网络攻击及防范
43. 怎么看网站的性能如何
44. 介绍http协议（特征）
45. 输入URL到页面加载显示完成发生了什么？
46. 说一下对cookie和session的认知，cookie有哪些限制？
47. 描述一下XSS和CRRSFI攻击？防御方法？
48. 知道304吗？什么时候用304？
49. 具体有哪些请求头是跟缓存相关的
50. Cookie有哪些字段可以设置
51. Cookie有哪些编码方式？
52. 前端优化策略
53. 除了cookie，还有什么存储方式。说说cookie和localstorage的区别
54. 浏览器输入网址到页面渲染全过程
55. HTML5和CSS3用的多吗？你了解他们的新属性吗？有在项目中用过吗？
56. Web性能优化
57. 浏览器缓存机制

### CSS

1. 说一下CSS盒模型
2. 画一条0.5px的线
3. Link标签和import标签的区别
4. Transition和animation的区别
5. Flex布局
6. BFC（块级格式化上下文，用于清除浮动，防止margin重叠等等）
7. 垂直居中的方法
8. 关于js动画CSS3动画的差异性
9. 说一下块元素和行元素
10. 多行元素的文本省略号
11. Visibility=hidden，opacity=0，display：none
12. 双边距重叠问题（外边距折叠）
13. Position，相关属性Position属性比较和区别
14. 浮动清除
15. CSS3新特性
16. CSS选择器有哪些，优先级呢
17. 怎么让一个元素消失
18. CSS动画如何实现
19. 如何实现图片在某个容器中居中的？
20. 如何实现元素的垂直居中
21. CSS3中对溢出的处理
22. Float的元素，display是什么
23. 隐藏页面中某个元素的方法
24. 三栏布局的实现方式，尽可能多写，浮动布局时，三个div的生成顺序有没有影响
25. Calc属性
26. 有一个width300，height300，怎么实现在屏幕上水平居中
27. Display：table和本身的table有什么区别
28. Z-index的定位方法
29. 如果想要改变一个dom元素的字体颜色，不在他本身上进行操作？
30. 对css的新属性的了解，用的最多的css属性是啥？
31. Line-height和height的区别
32. 设置一个元素的背景颜色，背景颜色会填充哪些区域？
33. 知道属性选择器和伪类选择器的优先级吗？
34. Inline-block、inline和block的区别，为什么img是inline还可以设置宽高
35. 用css实现一个硬币旋转的效果
36. 了解重绘和重排吗？知道怎么去减少重绘和重排吗？让文档脱离文档流有哪些方法
37. Css画正方体，三角形
38. Overflow的原理
39. Box-sizing的语法和基本用处
40. 使元素消失的方法有哪些
41. 两个嵌套的div，position都是absolute，子div设置top属性，那么这个top是相对于父元素的哪个位置定位的
42. Display
43. 怎么隐藏一个元素
44. Display：none和visibility：hidden的区别
45. 相对布局和绝对布局，position：relative和obsolute
46. Block、inline、inline-block的区别
47. Css的常用选择器
48. Css布局
49. Css定位
50. Relative定位规则
51. Css预处理器有什么

### JavaScript

1. Get请求传参长度的误区
2. 补充get和post请求在缓存方面的区别
3. 说一下闭包，闭包有什么用？用闭包写个单例模式
4. 说一下类的创建和继承
5. 如何解决异步回调地狱
6. 说说前端中的事件流
7. 如何让事件先冒泡后捕获
8. 说一下事件委托
9. 说一下图片的懒加载和预加载
10. Mouseover和mouseenter的区别
11. Js的new操作符做了哪些事情
12. 改变函数内部this指针的指向函数（bind，apply，call的区别）
13. Js的各种位置，比如clientheight，scrollheight，offsetheight以及scrolltop，offsettop，clienttop的区别
14. Js拖拽功能的实现
15. 异步加载js的方法
16. Ajax解决浏览器缓存问题
17. Js的节流和防抖
18. Js中的垃圾回收机制
19. Eval是做什么的
20. 如何理解前端模块化
21. 说一下commonjs，AMD和CMD
22. 对象深度克隆的简单实现
23. 实现一个once函数，传输函数参数只执行一次
24. 将原生的ajax封装成promise
25. Js监听对象属性的改变
26. 如何实现一个私有变量，用getName方法可以访问，不能直接访问
27. ==和===，以及object.js的区别
28. setTimeout，setInter和requestAnimationFrame之间的区别
29. 实现一个两列等高布局，讲讲思路
30. 自己实现一个bind函数
31. 用setTimeout来实现setInterval
32. Js怎么控制一次加载一直图片，加载完成在加载下一张
33. 代码的执行顺序
34. 如何实现sleep的效果（ES5或者ES6）
35. 简单的实现一个promise
36. Function.\_proto\_(getPrototypeOf)是什么？
37. 实现中所有对象的深度克隆（包装对象，date对象，正则对象）
38. 简单实现node的evens模块
39. 箭头函数this指向举例
40. Js判断类型
41. 数组常用方法
42. 数组去重
43. 时间代理在捕获阶段的实际应用
44. 去除字符串首尾空格
45. 性能优化
46. 能来讲讲js的语言特性吗？
47. 如何判断一个数组
48. 你说到typeof，能不能加一个限制条件达到判断条件
49. Js实现跨域
50. Js基本数据类型
51. Js的命名方式
52. Js深度拷贝一个元素的具体实现
53. 之前说了ES6set可以数组去重，是否还有数组去重的方法
54. Js的全排列
55. ES6中用过哪些
56. 跨域的原理
57. 不同数据类型的值的比较，是怎么转换的，有什么规则
58. Nul=defined为什么
59. This的指向哪几种
60. 暂停死区
61. AngularJS双向绑定原理
62. 写一个深度拷贝
63. 简历中提到了requestAnimationFrame，请问是怎么使用的
64. 有一个游戏叫做flappy Bird，就是一只小鸟在飞，前面是无尽的沙漠，上下不断有钢管生成，你要躲避钢管。然后小明在玩这个游戏时候老是卡顿甚至崩溃，说出原因（3-5个）已及解决办法
65. 编写代码，满足以下条件： （1）Hero("37er");执行结果为 Hi! This is 37er （2）Hero("37er").kill(1).recover(30);执行结果为 Hi! This is 37er Kill 1 bug Recover 30 bloods （3）Hero("37er").sleep(10).kill(2)执行结果为 Hi! This is 37er //等待10s后 Kill 2 bugs  //注意为bugs （双斜线后的为提示信息，不需要打印）
66. Jit；jc
67. ES6新特性
68. 说一下什么是virtual dom
69. Webpack用来干什么的
70. Ant-design优点和缺点
71. Js中继承实现的几种方式
72. 写一个函数，第一秒打印1，第二秒打印2
73. Vue的生命周期
74. 简单介绍一下symbol
75. 什么是事件监听
76. 介绍一下promise，及其底层是如实现的
77. 说说C++，java，JavaScript这三种语言
78. Js原型链，原型链是什么？Object的原型是什么？Object的原型的原型是什么？在数组原型链上实现删除数组重复数据的方法
79. Promise+Generator+Async的使用
80. 事件委托以及冒泡原理
81. 写个函数，可以转化下划线命名到驼峰命名
82. 深浅拷贝的区别和实现
83. Js中string的startwith和indexof两种方法的区别
84. Js字符串转字符的方法
85. Let const var的区别，什么是块级作用域，如何用ES5的方法实现块级作用域（立即执行函数），ES6呢
86. ES6箭头函数的特性
87. setTimeout和Promise的执行顺序
88. 有了解过事件模型吗？dom0和dom2级有什么区别，dom的分级是什么
89. 平时是怎么调试js的
90. Js的基本数据类型有哪些，基本数据类型和引用数据类型的区别，nan是什么的缩写，js的作用域类型，undefined==null返回的结果是什么，undefined与null的区别在哪，写一个函数判断变量类型
91. setTimeout（fn，100）；100毫秒是如何权衡的
92. js的垃圾回收机制
93. 写一个newbind函数，完成bind功能
94. 怎么获得对象上的属性：比如说通过object.key()
95. 简单说一下ES6的一些新特性
96. Call和apply是用来做什么？
97. 了解事件代理吗？这样做有什么好处
98. 如何使不同页面之间进行通信
99. 如何写一个继承
100. 给出以下代码，输出的结果是什么？原因？ for(var i=0;i<5;i++) { setTimeout(function(){ console.log(i); },1000); } console.log(i)
101. 给两个构造函数A和B，如何实现A继承B
102. 问能不能正常打印索引
103. 如果已经有三个promise，A、B、C，想串行执行，该怎么写？
104. 知道private和public吗
105. 基础的js
106. Async和await具体该怎么用？
107. 知道ES6、ES7的哪些语法
108. Promise和await/async的关系
109. Js的数据类型
110. Js加载过程阻塞，解决方法
111. Js对象类型，基本对象类型以及引用对象类型的区别
112. JavaScript中轮播实现原理？假如一个页面你上两个轮播，你会怎么实现？
113. 怎么实现一个计算一年中有多少周
114. 面向对象的继承
115. 引用类型常见的对象
116. ES6的常用
117. Class
118. 口述数组去重
119. ES6常用特性
120. 箭头函数和function有什么区别
121. New操作符原理
122. Bind，apply，call
123. Bind和apply的区别
124. 数组的去重
125. Assign的深拷贝
126. 说promise，没有promise怎么办
127. 事件委托
128. 怎么用原生的js实现jQuery的一个特定方法
129. Arguments
130. 箭头函数获取arguments
131. 模块化开发
132. Eventloop

### jQuery

1. .jquery源代码
2. jQuery的一个方法的实现原理

bootstrap

1. bootstrap清除浮动的方法
2. 请你说一下bootstrap

### 前端核心———移动web开发

1. 移动应用和web应用的关系

答：一、地位不同

web属于一种技术，而webapp则是根据这一技术开发成的应用，web是源头，而webapp则是支流。

二、功能不同

web app特指的是用html5技术开发，之所以叫web app是因为他比较接近客户端应用程序的用户体验，可以和系统深度融合，调用一些只有客户端才能调用的功能（比如在移动设备上利用html5开发出的网页可以访问电话、摄像头等本地功能）。

三、功能不同

移动Web App跨平台开发，用户不需要去卖场来下载安装App，任何时候都可以发布App，因为根本不需要官方卖场的审核。web直接下载到设备，以独立的应用程序运行(并不需要浏览器)，用户必须手动去下载并安装这些原生App。

四、版本控制的区别

移动Web App，所有的用户都是用同样的版本。web用户可以自由地选择是否更新软件版本，所以会出现不同用户同时使用不同版本的情况。

前端核心———ajax

1. Ajax返回的状态
2. 实现一个ajax
3. 如何实现ajax请求，假如我有过个请求权，我需要让这些ajax请求按照某种顺序依次执行，有什么方法呢？如何处理ajax跨域
4. 写出原生ajax

### 前端核心———json

1. Jsonp的缺点
2. 跨域（jsonp、ajax）
3. 如何实现跨域
4. Dom是什么？你的理解
5. 关于dom的api有什么

### 前端基础———bootstrap

1. Bootstrap清除浮动的方法
2. 说说bootstrap

### Vue

1. 谈谈你对MVVM开发模式的理解

MVVM分为Model、View、ViewModel三者。

Model 代表数据模型，数据和业务逻辑都在Model层中定义；

View 代表UI视图，负责数据的展示；

ViewModel 负责监听 Model 中数据的改变并且控制视图的更新，处理用户交互操作；

Model 和 View 并无直接关联，而是通过 ViewModel 来进行联系的，Model 和 ViewModel 之间有着双向数据绑定的联系。因此当 Model 中的数据改变时会触发 View 层的刷新，View 中由于用户交互操作而改变的数据也会在 Model 中同步。

这种模式实现了 Model 和 View 的数据自动同步，因此开发者只需要专注对数据的维护操作即可，而不需要自己操作 dom。

1. Vue 有哪些指令？

v-html、v-show、v-if、v-for等等

1. v-if 和 v-show 有什么区别？

v-show 仅仅控制元素的显示方式，将 display 属性在 block 和 none 来回切换；而v-if会控制这个 DOM 节点的存在与否。当我们需要经常切换某个元素的显示/隐藏时，使用v-show会更加节省性能上的开销；当只需要一次显示或隐藏时，使用v-if更加合理。

1. 简述Vue的响应式原理

当一个Vue实例创建时，vue会遍历data选项的属性，用 Object.defineProperty 将它们转为 getter/setter并且在内部追踪相关依赖，在属性被访问和修改时通知变化。

每个组件实例都有相应的 watcher 程序实例，它会在组件渲染的过程中把属性记录为依赖，之后当依赖项的 setter 被调用时，会通知 watcher 重新计算，从而致使它关联的组件得以更新。

<https://www.cnblogs.com/chenjg/p/9541291.html>

1. Vue中如何在组件内部实现一个双向数据绑定？

假设有一个输入框组件，用户输入时，同步父组件页面中的数据

具体思路：父组件通过 props 传值给子组件，子组件通过 $emit 来通知父组件修改相应的props值，具体实现如下：

import Vue from 'vue'

const component = {

props: ['value'],

template: `

<div>

<input type="text" @input="handleInput" :value="value">

</div>

`,

data () {

return {

}

},

methods: {

handleInput (e) {

this.$emit('input', e.target.value)

}

}

}

new Vue({

components: {

CompOne: component

},

el: '#root',

template: `

<div>

<comp-one :value1="value" @input="value = arguments[0]"></comp-one>

</div>

`,

data () {

return {

value: '123'

}

}

})

可以看到，当输入数据时，父子组件中的数据是同步改变的：

Image 1.png

Image 2.png

我们在父组件中做了两件事，一是给子组件传入props，二是监听input事件并同步自己的value属性。那么这两步操作能否再精简一下呢？答案是可以的，你只需要修改父组件：

template: `

<div>

<!--<comp-one :value1="value" @input="value = arguments[0]"></comp-one>-->

<comp-one v-model="value"></comp-one>

</div>

`

v-model 实际上会帮我们完成上面的两步操作。

1. Vue中如何监控某个属性值的变化？

比如现在需要监控data中，obj.a 的变化。Vue中监控对象属性的变化你可以这样：

watch: {

obj: {

handler (newValue, oldValue) {

console.log('obj changed')

},

deep: true

}

}

deep属性表示深层遍历，但是这么写会监控obj的所有属性变化，并不是我们想要的效果，所以做点修改：

watch: {

'obj.a': {

handler (newName, oldName) {

console.log('obj.a changed')

}

}

}

还有一种方法，可以通过computed 来实现，只需要：

computed: {

a1 () {

return this.obj.a

}

}

利用计算属性的特性来实现，当依赖改变时，便会重新计算一个新值。

1. Vue中给data中的对象属性添加一个新的属性时会发生什么，如何解决？

示例：

<template>

<div>

<ul>

<li v-for="value in obj" :key="value">

{{value}}

</li>

</ul>

<button @click="addObjB">添加obj.b</button>

</div>

</template>

<script>

export default {

data () {

return {

obj: {

a: 'obj.a'

}

}

},

methods: {

addObjB () {

this.obj.b = 'obj.b'

console.log(this.obj)

}

}

}

</script>

<style></style>

点击button会发现，obj.b 已经成功添加，但是视图并未刷新：

Image 3.png

Image 4.png

原因在于在Vue实例创建时，obj.b并未声明，因此就没有被Vue转换为响应式的属性，自然就不会触发视图的更新，这时就需要使用Vue的全局api $set()：

addObjB () {

// this.obj.b = 'obj.b'

this.$set(this.obj, 'b', 'obj.b')

console.log(this.obj)

}

$set()方法相当于手动的去把obj.b处理成一个响应式的属性，此时视图也会跟着改变了：

Image 5.png

1. delete和Vue.delete删除数组的区别

delete只是被删除的元素变成了 empty/undefined 其他的元素的键值还是不变。

Vue.delete直接删除了数组 改变了数组的键值。

var a=[1,2,3,4]

var b=[1,2,3,4]

delete a[1]

console.log(a)

this.$delete(b,1)

console.log(b)

近期面试题汇总

### 滴滴出行

1. ul中有大量li，如何实现点击一个li则显示li中的内容
2. 页面优化
3. js的兼容性问题

<https://www.cnblogs.com/kingwu/p/9556506.html>

<https://www.cnblogs.com/JackSmall/p/8516840.html>

1. on与onclick的区别

<https://blog.csdn.net/qq_29715077/article/details/80013893>

简单的说页面加载完成时候页面显示的元素（DOM节点已全部加载完）可以用 on ， 也可以用click ，

但是在页面加载完成之后后期再追加元素（DOM节点元素还没完全显示出来）只能用on

亮亮视野科技

1. localstorage与sessionstorage
2. 不同浏览器内核样式兼容

<https://blog.csdn.net/chenzlyan/article/details/80567167>

1. 描述ajax过程

<https://blog.csdn.net/hlinghling/article/details/52624757>

### 北森

1. 数组中有大量元素，如何找出数量最多的元素及其个数

<https://blog.csdn.net/weixin_40539125/article/details/82880108>

1. 盒模型
2. px,em,rem的区别

<https://www.runoob.com/w3cnote/px-em-rem-different.html>

1. hash与map的区别

<https://www.cnblogs.com/inception6-lxc/p/9263964.html>

1. 表格中有大量单元格，如何实现点击一个单元格则显示单元格中的内容
2. js继承

<https://www.cnblogs.com/Grace-zyy/p/8206002.html>

1. for循环中的settimeout问题

<https://blog.csdn.net/yihanzhi/article/details/79565628>

<https://blog.csdn.net/gaos2310/article/details/78411388>

1. es6新增数据类型是什么，有什么作用

<https://www.jianshu.com/p/61da27746d9a>

1. 如何交换dom树上的两个节点

<https://segmentfault.com/q/1010000001691234>

<https://www.jianshu.com/p/56ae40543587>

### 薪人薪事

1. 基本数据类型有哪些

<https://blog.csdn.net/qq_35630674/article/details/83089711>

1. css选择器优先级（id、类、属性....）

<https://www.cnblogs.com/zouyitangzaishuo/p/5265563.html>

<https://blog.csdn.net/Summer_15/article/details/70216187>

1. 盒模型的互相转换

<https://www.cnblogs.com/huangzhilong/p/5007157.html>

<https://www.cnblogs.com/Jm-jing/articles/7049239.html>

1. ul中有大量li，如何实现点击一个li则显示li中的内容

<https://blog.csdn.net/u010513603/article/details/80214688>

<https://blog.csdn.net/R_banyan0820/article/details/82865347>

1. 什么是事件代理（手写代码）

<https://www.jianshu.com/p/a77d8928c5c9>

<https://blog.csdn.net/weixin_43647163/article/details/86776609>

1. js继承（手写代码）
2. 两栏布局，固定左侧div宽度，如何让右侧div填充剩余空间

<https://blog.csdn.net/qq_31965515/article/details/81394516>

1. position有哪些

<https://blog.csdn.net/HHH_LLL/article/details/88765208>

<https://zhidao.baidu.com/question/524525639354660925.html>

1. 字符串从后往前每三位加一个逗号如何实现（手写代码）

<https://submaze.iteye.com/blog/1866129>

1. flex布局

<http://www.ruanyifeng.com/blog/2018/10/flexbox-form.html>

<https://blog.csdn.net/lm9948/article/details/80732702>

1. vue中v-if与v-show的区别
2. 冒泡与捕获

<https://www.cnblogs.com/christineqing/p/7607113.html>

<https://www.cnblogs.com/msdn1433/p/5666844.html>

1. var undifined 1==true 2==true 0==false null==undefined []==![] 0==''写出结果

<https://blog.csdn.net/qq_18416057/article/details/70155393>

1. 不使用sort函数对数组数字进行排序（手写代码）

<https://www.cnblogs.com/7qin/p/10197943.html>

<https://blog.csdn.net/Luyanc/article/details/93169195>

<https://www.cnblogs.com/miaoqing/p/5586995.html>

1. 半透明弹出框实现垂直居中，至少一种（手写代码）

<http://www.softwhy.com/article-9126-1.html>

<https://blog.csdn.net/xyw_blog/article/details/21483529>

1. 为什么字符串/数字可以调用方法

<https://blog.csdn.net/baidu_33736953/article/details/85319859>

1. 如何获取html中所有节点并显示各节点种类与对应数量

<https://blog.csdn.net/maid_04/article/details/80002258>

1. 如何判断数据类型

<https://www.cnblogs.com/yi0921/p/6183422.html>

1. 如何判断是否为数组

<https://www.cnblogs.com/echolun/p/10287616.html>

1. typeof数组结果是什么

<https://www.cnblogs.com/yanze/p/5988217.html>

1. typeof函数结果是什么

<https://blog.csdn.net/weixin_38362146/article/details/78036252>

<https://zhidao.baidu.com/question/79159257.html>

1. 解释一下什么是原型

<https://blog.csdn.net/qq_42019025/article/details/80708446>

1. 字符串如何转变为数组

<https://www.cnblogs.com/sese/p/9802780.html>

<https://blog.csdn.net/weixin_37744601/article/details/82291396>

1. dom操作（获取）

<https://blog.csdn.net/shuidinaozhongyan/article/details/77825000>

1. 数组涉及的方法

<https://www.cnblogs.com/sherryweb/p/11024141.html>

1. vue生命周期

<https://blog.csdn.net/mqingo/article/details/86031260>

1. 深浅拷贝

<https://www.cnblogs.com/136asdxxl/p/8645750.html>

### 海天起点

1. V-if、v-show的区别
2. 前端页面三层

<https://blog.csdn.net/caowen414/article/details/42837265>

1. rgba和opacity区别

<https://www.jianshu.com/p/74288368da98>

opacity会继承父元素的opacity 属性，而RGBA设置的元素的后代元素不会继承不透明属性。简单来说就是opacity作用于元素和元素所有内容的透明

rgba相对于opacity还是技高一筹的，当然只要是涉及颜色的，都可以用rgba来设置

1. display：none和visibility：hidden有什么区别

display:none是个相当惨无人道的声明，子孙后代全部搞死（株连性），而且连块安葬的地方都不留（不留空间），导致全体民众哗然（渲染与回流）。 visibility:hidden则具有人道主义关怀，虽然不得已搞死子孙，但是子孙可以通过一定手段避免（伪株连性），而且死后全尸，墓地俱全（占据空间），国内民众比较淡然（无渲染与回流）

1. 主流前端框架
2. 前端优化
3. Alert取消

<https://www.jb51.net/article/149591.htm>

confirm？

### 布本智能

1. Vue和jQuery的区别

<http://m.php.cn/article/412970.html>

1. Vue生命周期

<https://www.cnblogs.com/zdz8207/p/vue-lifecycle.html>

1. Web存储
2. jquery链式编程

<https://blog.csdn.net/qq_43294510/article/details/88563264>