1. Bbbbb0-0---0---Promise 捕获异常 async

<https://blog.csdn.net/liwusen/article/details/79617903>

1. var let 区别 let暂时性死区
2. Git

第一步：$> ssh一keygen 一t rss 一C zhangsan@abc.com

第二步：拷贝公钥到gitlab

第三步：

$> git config —global user.name zhangsan

$> git config —global user.email zhangsan@abc.com

第四步：$> git clone git@lab.com:org/project.git

第五步：$> git checkout 一b project一20170227一zhangsan一bugfix

第六步：修改代码

第七步：git status

第八步：git add .

第九步：git commit 一am ‘bugfix’

第十步：

git push 一一set一upstream origin project一20170227一zhangsan一bugfix

1. Webpack的配置：React里md文档第一天webpack
2. 浏览器比如说谷歌的最小字体是多大。如何解决

12px,

.small-font{

　　　  font-size:12px;

            -webkit-transform-origin-x: 0;

            -webkit-transform: scale(0.90);

        }

        .smallsize-font {

 　　　　font-size:10.8px;

       　　        }

　　<p style="color: #FF0000;" class="small-font smallsize-font">温馨提示：</p>

1. 适配移动端的方法

<https://blog.csdn.net/chenjuan1993/article/details/81710022>

1. Ssr

服务端渲染（SSR）与客户端渲染（BSR）都是数据渲染的主要的方式

客户端渲染:前端利用ajax等数据交互手段获取服务端提供的数据之后，渲染到HTML页面。

服务器端渲染:整个网站先在服务器中运行，然后返回一个完整的HTML字符串，将这个字符串当成响应内容输出给浏览器。

SSR优势

利于搜索引擎抓取我们的页面。

build之后，会静态化page页面，所以访问速度快。

1. Nuxt:md文档14
2. 跨域 除了jsonp还有什么

jsonp、 iframe、window.name、window.postMessage、服务器上设置代理页面

1. 数组方法 ，字符串方法:二阶段笔记
2. 数组深拷贝浅拷贝。

i.浅拷贝深拷贝

//拷贝就是把父对象的属性，全部拷贝给子对象。

var Chinese = {

nation:'中国’

}

Var Doctor={

career:'医生'

}

function extendCopy(p) {

var c = {};

for (var i in p) {

c[i] = p[i];

}

c.uber = p; return c;

}

//使用的时候，这样写：

Doctor = extendCopy(Chinese);

Doctor.career ='医生'；

alert(Doctor.nation); // 中国

//但是，这样的拷贝有一个问题。那就是，如果父对象的属性等于数组或另一个对象，那么实际上，子对象获得的只是一个内存地址，而不是真正拷贝，因此存在父对象被篡改的可能。

//请看，现在给Chinese添加一个"出生地"属性，它的值是一个数组。

Chinese.birthPlaces =['北京','上海','香港'];

//通过extendCopy()函数，Doctor继承了Chinese。

Doctor = extendCopy(Chinese);

//然后，我们为Doctor的"出生地"添加一个城市：

Doctor.birthPlaces.push('厦门')；

//看一下输入结果

alert(Doctor.birthPlaces); //北京，上海，香港，厦门 alert(Chinese.birthPlaces); //北京，上海，香港，厦门

//结果是两个的出生地都被改了。

//所以，extendCopy()只是拷贝了基本类型的数据，我们把这种拷贝叫做''浅拷贝〃。

2.深拷贝

//所谓"深拷贝"，就是能够实现真正意义上的数组和对象的拷贝。它的实现并不难， 只要递归调用"浅拷贝"就行了。

var Chinese = {

nation:'中国’

}

var Doctor = {

career:'医生'

}

function deepCopy(p, c) {

var c = c || {};

for (var i in p) {

if (typeof p[i] === 'object') {

c[i] = (p[i].constructor === Array) ? [] : {}; deepCopy(p[i], c[i]);

} else {

c[i] = p[i];

}

}

return c;

}

//看一下使用方法：

Doctor = deepCopy(Chinese);

//现在，给父对象加一个属性，值为数组。然后，在子对象上修改这个属性：

Chinese.birthPlaces =['北京','上海','香港'];

Doctor.birthPlaces.push('厦门')；

alert(Doctor.birthPlaces); //北京，上海，香港，厦门

alert(Chinese.birthPlaces); //北京，上海，香港

JavaScript中的对象一般是可变的（Mutable),因为使用了引用赋值，新的对象简单的引用了原始对象，改变新的对象将影响到原始对象。如'foo={a: 1}; bar=foo; bar.a=2'你会发现此时 'foo.a'也被改成了 '2'。

虽然这样做可以节约内存，但当应用复杂后，这就造成了非常大的隐患，Mutable带来的优点变得得不偿失。为了解决这个问题，一般的做法是使用shallowCopy (浅拷贝）或deepCopy (深拷贝）来避免被修改，但这样做造成了CPU和内存的浪费。

Immutable可以很好地解决这些问题。（同时见二阶段笔记）

1. React生命周期,新增的，废除的，
2. 事件委托，事件（见二阶段笔记）
3. Redux，React-Redux connect原理，reducers纯函数
4. map和forEach区别

<https://blog.csdn.net/zgrkaka/article/details/79650804>

1. Vue.config.js配置

<https://segmentfault.com/a/1190000015801427>



1. Render什么时候渲染

setstate之后<https://blog.csdn.net/fengtingyan/article/details/83026387>

1. 高阶组件
2. setState同步还是异步

异步

1. 输入url到页面加载完成

分为4个步骤：

1. 当发送一个 URL 请求时，不管这个 URL 是 Web 页面的 URL 还是 Web 页面上每个资源的 URL，浏览器都会开启一个线程来处理这个请求，同时在远程 DNS 服务器上启动一个 DNS 查询。这能使浏览器获得请求对应的 IP 地址。

2. 浏览器与远程 Web 服务器通过 TCP 三次握手协商来建立一个 TCP/IP 连接。该握手包括一个同步报文，一个同步-应答报文和一个应答报文，这三个报文在 浏览器和服务器之间传递。该握手首先由客户端尝试建立起通信，而后服务器应答并接受客户端的请求，最后由客户端发出该请求已经被接受的报文。

3. 一旦 TCP/IP 连接建立，浏览器会通过该连接向远程服务器发送 HTTP 的 GET 请求。远程服务器找到资源并使用 HTTP 响应返回该资源，值为 200 的 HTTP 响应状态表示一个正确的响应。

4. 此时，Web 服务器提供资源服务，客户端开始下载资源。

21.插槽

22.redux里面怎么写异步

23.ES6新增了哪些特性

http://es6.ruanyifeng.com/

1)块作用域一let

2)衡量一const

3)解构数组一Array Destructuring

4)解构对象一Object Destructuring

5)模板字符串一Template Strings

6)展开操作符

7)剩余操作符

8)解构参数

9)箭头函数

10)对象表达式

11)对象属性名

12)对比两个值是否相等

13)把对象的值复制到另一个对象里

14)设置对象的prototype

15)\_\_proto\_\_

16)supper

17)迭代器

18)class 类

19)get set

20)静态方法

21)继承

22)模块化

24.React组件传值

25.Null和undefined区别

<https://www.cnblogs.com/zhaoxinmei-123/p/9046962.html>

<https://zhidao.baidu.com/question/336242397.html?qq-pf-to=pcqq.c2c>

1. 小公司的jquery

<https://www.w3school.com.cn/jquery/index.asp>

1. 跨域的原理、jsonp原理

<https://baijiahao.baidu.com/s?id=1597142798213225251&wfr=spider&for=pc&qq-pf-to=pcqq.c2c>

1. promise原理

<https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Promise>

1. js封装jq改变css的方法

<div id='app'>hello</div>

Object.prototype.css = function(obj){

for( var key in obj){

this.style[key] = obj[key];

}

}

app.css({'background':'red', 'color':'white'})

1. 路由跳转原理

利用 URL 的 hash(#)

利用 H5 新增方法 History interface

<https://www.cnblogs.com/liangyin/p/9686696.html>

1. 大公司 react

<https://react.docschina.org/docs/hello-world.html>

1. 表单验证

<https://cn.vuejs.org/v2/guide/forms.html>

1. elementUI

[https://element.eleme.cn/#/zh-CN/component/installation](https://element.eleme.cn/" \l "/zh-CN/component/installation)

1. vue创建组件

组件：<https://cn.vuejs.org/v2/guide/single-file-components.html>

插件：<https://cn.vuejs.org/v2/guide/plugins.html>

1. vue双向绑定原理，几层原理

<https://cn.vuejs.org/v2/guide/reactivity.html>

1. react 新增的生命周期

<https://zh-hans.reactjs.org/docs/react-component.html>

static getDerivedStateFromProps()

getSnapshotBeforeUpdate()

1. vue和react的区别

[https://cn.vuejs.org/v2/guide/comparison.html#React](https://cn.vuejs.org/v2/guide/comparison.html" \l "React)

1. vue中的ref

[https://cn.vuejs.org/v2/guide/components-edge-cases.html#%E8%AE%BF%E9%97%AE%E5%AD%90%E7%BB%84%E4%BB%B6%E5%AE%9E%E4%BE%8B%E6%88%96%E5%AD%90%E5%85%83%E7%B4%A0](https://cn.vuejs.org/v2/guide/components-edge-cases.html" \l "%E8%AE%BF%E9%97%AE%E5%AD%90%E7%BB%84%E4%BB%B6%E5%AE%9E%E4%BE%8B%E6%88%96%E5%AD%90%E5%85%83%E7%B4%A0)

ref就是一个标记

<input ref="abc" />

this.$refs.abc === input

1. 面试直接问的vue，路由 组件，钩子函数，路由传参，vuex原理 项目里的细节，组件数据传递方式
2. Redux中如何发送异步请求

41.css实现三角形（见一阶段笔记）

42判断数组有哪些方法

方法一.判断其是否具有“数组性质”，如slice()方法。可自己给该变量定义slice方法，故有时会失效

方法二.obj instanceof Array 在某些IE版本中不正确

方法三.方法一二皆有漏洞，在ECMA Script5中定义了新方法Array.isArray(), 保证其兼容性，最好的方法如下：

toString.call(18);//”[object Number]”

toString.call(“”);//”[object String]”

解析这种简单的数据类型直接通过typeof就可以直接判断

toString.call常用于判断数组、正则这些复杂类型

toString.call(/[0-9]{10}/)//”[object RegExp]”

if(typeof Array.isArray==="undefined"){

Array.isArray = function(arg){

return Object.prototype.toString.call(arg)==="[object Array]"

};

}

1. vue里wacth有哪些属性：getter，setter
2. vue 如何实现懒加载(lazy)

<https://blog.csdn.net/w1366352655/article/details/85855821>

45.箭头函数与function在运用中有哪些区别

<https://segmentfault.com/a/1190000018477740?utm_source=tag-newest>

<http://www.softwhy.com/article-9330-1.html>

1. 如何设计节流函数（思路：在规定时间内只触发一次）

<https://www.cnblogs.com/zhangycun/p/10949450.html>

1. vuex的底层原理是啥

<http://www.sohu.com/a/290830922_505779?qq-pf-to=pcqq.c2c>

1. webpack自己会搭建吗
2. es6的几个特别重要，大家一定要看下es6最好懂原理，还有就是跨域几种方式的原理，还有redux原理，react的到底是个什么
3. 项目你最近项目是什么项目，你做了什么，你项目里面有啥棘手问题吗，出现什么bug，你当时用了多久时间怎么解决的
4. ajax原理（二阶段笔记）
5. jsonp原理（二阶段笔记）
6. 闭包原理（二阶段笔记）
7. 生成器函数和await async的区别

<http://www.bubuko.com/infodetail-2037441.html>

<https://blog.csdn.net/qq_38021852/article/details/82778151>

<https://blog.csdn.net/deng1456694385/article/details/83831931>

1. async/await和promise如何结合使用

<https://blog.csdn.net/besttoby01/article/details/90517198>

55.浏览器缓存机制是什么

<https://blog.csdn.net/WDCCSDN/article/details/82498413>

1. 事件循环机制

<https://www.cnblogs.com/gwf93/p/10718292.html>

1. vue和react的区别
2. 数据双向绑定用在什么地方（除了表单还有什么） 原理

下拉刷新，底部加载更多，如果没有会显示出没有数据等，提交表单+三阶段笔记

1. 如何防止XSS攻击

(1)将前端输出数据都进行转义

(2)将输出的字符串中的\反斜杠进行转义

(3)从url中获取的信息，防止方法是由后端获取，在前端转义后再行输出

(4)使用cookie的HttpOnly属性，保护好cookie

1. 项目开发流程 ：开始前 项目经理会开会 展示项目ui图 确定项目周期 分配任务 然后刚开始 前端写模板，静态页面 后端开发接口 经常会开会确认个人进度 分配任务 看代码质量 后端开发好接口后 前后端协同开发 后端给的地址一般是这样
2. 随机数Math.random()可以取到0和1吗？（二阶段笔记）
3. 如何打乱数组？

<https://blog.csdn.net/m0_38082783/article/details/79579116>

1. 解决v-model和vuex的冲突

computed getter、setter

<https://blog.csdn.net/weixin_33824363/article/details/91434500>

1. vue中的数据改变，怎么调试，知道哪里改变？（不是组件内部）

vue-devtools 工具，就是做调试用的，里面能看到相关的数据

1. 饿了么ui和蚂蚁金服区别是什么？

<https://www.zhihu.com/question/65498836>

1. vue中的点穿事件遇过吗？怎么处理

<https://www.cnblogs.com/panyujun/p/11065741.html>

1. rem怎么实现？

<https://blog.csdn.net/smalljiayi/article/details/79044902>

1. 如何屏蔽关键字？

实现类似 replace 的效果，就是查找到关键字，替换成空或xx

1. 有自己开发的公共组件吗

就是插件，plugin，我们做完后提交到网上，别人npm install 下载使用的

1. promise中，在return之前是异步还是同步？

没读懂问题，代码执行顺序查看 eventloop、宏任务、微任务

1. vue-router 路由中的中hash模式和history模式的区别

hash模式URL永远带#号，history不含#，history更加利于推广宣传，但是在history模式下进行刷新会出现404错误，需要后端配置重定向到路由，hash模式刷新会到对应页面

1. less和sass的区别，为啥要使用sass

<https://blog.csdn.net/pedrojuliet/article/details/72887490>

两个都属于css预处理 less是客户端的渲染，可以直接引入js使用 sass是服务端做处理

1. post会请求两次，两次是什么

<https://www.jianshu.com/p/ee34d81df0d1>

<https://blog.csdn.net/seanxwq/article/details/83027263>

<https://www.cnblogs.com/CarrotHu/p/10499838.html>

1. git stash命令的作用就是将目前还不想提交的但是已经修改的内容进行保存至堆栈中，后续可以在某个分支上恢复出堆栈中的内容。这也就是说，stash中的内容不仅仅可以恢复到原先开发的分支，也可以恢复到其他任意指定的分支上。git stash作用的范围包括工作区和暂存区中的内容，也就是说没有提交的内容都会保存至堆栈中。
2. Rem是如何实现自适应布局的？

<https://blog.csdn.net/smalljiayi/article/details/79044902>

笔试题

1.代码输入依次为：

setTimeout(function(){

console.log(1)

})

new Promise(function(resolve){

console.log(2)

resolve(3)

}).then(function(val){

console.log(val)

})

console.log(4)

1. 防抖与节流的区别

<https://www.jianshu.com/p/51ce886fb7f6>

1. 如何实现浏览器内多个标签页之间的通信

通过WebSocket或SharedWorker把客户端和服务器端建立socket连接，从而实现通信；也可以调用localstorge、cookies等本地存储方法。

1. 请手写一个原型链继承

5将字符串相邻相同的分别输出，比如“aabbccabttttr”输出 aa bb cc a b tttt r

<https://blog.csdn.net/A123333333333/article/details/83246802>

1. 如何发送同步非缓存的ajax？

https://blog.csdn.net/umedier/article/details/82455109

1. 数组去重，要求：在原型链上添加函数

1.JS数组去重

以下是展示三种方法：

Array.prototype.unique1 = function () {

var n = []; //一个新的临时数组

for (var i = 0; i < this.length; i++) //遍历当前数组

{

//如果当前数组的第i已经保存进了临时数组，那么跳过，

//否则把当前项push到临时数组里面

if (n.indexOf(this[i]) == -1) n.push(this[i]);

}

return n;

}

Array.prototype.unique2 = function()

{

var n = {},r=[]; //n为hash表，r为临时数组

for(var i = 0; i < this.length; i++) //遍历当前数组

{

if (!n[this[i]]) //如果hash表中没有当前项

{

n[this[i]] = true; //存入hash表

r.push(this[i]); //把当前数组的当前项push到临时数组里面

}

}

return r;

}

Array.prototype.unique3 = function()

{

var n = [this[0]]; //结果数组

for(var i = 1; i < this.length; i++) //从第二项开始遍历

{

//如果当前数组的第i项在当前数组中第一次出现的位置不是i，

//那么表示第i项是重复的，忽略掉。否则存入结果数组

if (this.indexOf(this[i]) == i) n.push(this[i]);

}

return n;

}

1. 原生js怎么获取url中的内容window.location.href
2. Vue.js是一个提供MVVM数据双向绑定的库，专注于UI层面，核心思想是：数据驱动、组件系统。
3. 请分别写出 windows 和 linux 操作系统中查看 ip 的命令

<https://blog.csdn.net/wd2011063437/article/details/80164380>

10vue-devtools

vue里面为什么不用jq

Vue 的官方是不建议直接操作 DOM 的，Vue 的用途在于视图和数据的绑定。如果通过JQuery 直接操作 DOM 的话，势必会造成视图数据和模型数据的不匹配，这样 Vue 就失去它存在的意义了。  
　　JQuery 和 VueJS 合理使用并不会造成冲突，因为他们的侧重点不同，VueJS 侧重数据绑定和视图组件，JQuery 侧重异步请求和动画效果。如果使用JQuery + VueJS 开发，一定要在 Vue 渲染完所有的 HTML组件之后再通过 JQuery 处理，而使用 JQuery 时应避免直接操作 DOM ，但是应用动画是允许的。

1. 一个 npm 包里的 package.json 具备的必要字段有哪些

<https://blog.csdn.net/weixin_43234831/article/details/91054103>

11实现 add 函数 add(1)(2)(3) // 6  
  
给定两个数组，写一个方法来计算他们的交集。 例如，给定 nums1 = [1,2,2,1] nums2 = [2,2]，要求返回 [2,2]  
  
实现 (5).add(5).minus(1)功能，结果为9

1. display:none和visibility:hidden的区别就是：  
     
   display:none：隐藏一个元素，占据的空间会消失  
     
   visibility:hidden：使一个元素开不见，但是占据的空间不会消失，常用在布局里面
2. css引入的方式有哪些，link和@import的区别是什么  
   有四种形式:  
   1.链入外部样式表,就是把样式表保存为一个样式表文件,然后在页面中用<link rel="stylesheet" type="text/css" href="\*.css">链接这个样式表文件.  
   2.内部样式表,就是把样式表放到页面的<head>区里. <style type="text/css">  
   div {height: 600px;}  
   </style>  
   3.导入外部样式表,用@import,在<head>与</head>之间，<style type="text/css">  
   <!一一  
   @import "\*.css"  
   一一>  
   </style>  
   4.内嵌样式,就是在标签内写入style="",比如:  
   <div style="background:#cccccc"></div>设置div背景色为灰色.  
   区别：  
   1）. link是XHTML标签，除了加载CSS外，还可以定义RSS等其他事务；@import属于CSS范畴，只能加载CSS。  
   2）. link引用CSS时，在页面载入时同时加载；@import需要页面网页完全载入以后加载。  
   3）. link是XHTML标签，无兼容问题；@import是在CSS2.1提出的，低版本的浏览器不支持。  
   4）. link支持使用Javascript控制DOM去改变样式；而@import不支持。
3. CSS 选择符有哪些？哪些属性可以继承？优先级算法如何计算？ CSS3新增伪类有那些？
4. \* 1.id选择器（ # myid）
5. 2.类选择器（.myclassname）
6. 3.标签选择器（div, h1, p）
7. 4.相邻选择器（h1 + p）
8. 5.子选择器（ul < li）
9. 6.后代选择器（li a）
10. 7.通配符选择器（ \* ）
11. 8.属性选择器（a[rel = "external"]）
12. 9.伪类选择器（a: hover, li: nth - child）
13. \* 可继承： font-size font-family color, UL LI DL DD DT;
14. \* 不可继承 ：border padding margin width height ;
15. \* 优先级就近原则，样式定义最近者为准;
16. \* 载入样式以最后载入的定位为准;
17. 优先级为:
18. !important > id > class > tag
19. important 比 内联优先级高
20. CSS3新增伪类举例：
21. p:first-of-type 选择属于其父元素的首个 <p> 元素的每个 <p> 元素。
22. p:last-of-type 选择属于其父元素的最后 <p> 元素的每个 <p> 元素。
23. p:only-of-type 选择属于其父元素唯一的 <p> 元素的每个 <p> 元素。
24. p:only-child 选择属于其父元素的唯一子元素的每个 <p> 元素。
25. p:nth-child(2) 选择属于其父元素的第二个子元素的每个 <p> 元素。
26. :enabled、:disabled 控制表单控件的禁用状态。

:checked，单选框或复选框被选中。

13

.为什么虚拟dom提高性能？

1. web应用从服务器主动推送 data 到客户端的方法？ socket
2. 事件传播的三个阶段
3. Html5中本地存储概念是什么，有什么优点，与cookie有什么区别？  
   html5中的Web Storage包括了两种存储方式：sessionStorage和localStorage。  
   sessionStorage用于本地存储一个会话（session）中的数据，这些数据只有在同一个会话中的页面才能访问并且当会话结束后数据也随之销毁。因此sessionStorage不是一种持久化的本地存储，仅仅是会话级别的存储。而localStorage用于持久化的本地存储，除非主动删除数据，否则数据是永远不会过期的；  
   cookie是网站为了标示用户身份而储存在用户本地终端（Client Side）上的数据（通常经过加密）。  
   区别：  
   1、cookie数据始终在同源的http请求中携带（即使不需要），即cookie在浏览器和服务器间来回传递。而sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。cookie数据还有路径（path）的概念，可以限制cookie只属于某个路径下。  
   2、存储大小限制也不同，cookie数据不能超过4k，同时因为每次http请求都会携带cookie，所以cookie只适合保存很小的数据，如会话标识。sessionStorage和localStorage 虽然也有存储大小的限制，但比cookie大得多，可以达到5M或更大。  
   3、数据有效期不同，sessionStorage：仅在当前浏览器窗口关闭前有效，自然也就不可能持久保持；localStorage：始终有效，窗口或浏览器关闭也一直保存，因此用作持久数据；cookie只在设置的cookie过期时间之前一直有效，即使窗口或浏览器关闭。  
   4、作用域不同，sessionStorage不在不同的浏览器窗口中共享，即使是同一个页面；localStorage 在所有同源窗口中都是共享的；cookie也是在所有同源窗口中都是共享的。
4. 清除浮动有哪些方式？比较好的方式是哪一种？  
   (Q1)  
   （1）父级div定义height。  
   （2）结尾处加空div标签clear:both。  
   （3）父级div定义伪类:after和zoom。  
   （4）父级div定义overflow:hidden。  
   （5）父级div定义overflow:auto。  
   （6）父级div也浮动，需要定义宽度。  
   （7）父级div定义display:table。  
   （8）结尾处加br标签clear:both。  
   (Q2)比较好的是第3种方式，好多网站都这么用。
5. 你有哪些性能优化的方法？  
   （1） 减少http请求次数：CSS Sprites, JS、CSS源码压缩、图片大小控制合适；网页Gzip，CDN托管，data缓存 ，图片服务器。  
   （2） 前端模板 JS+数据，减少由于HTML标签导致的带宽浪费，前端用变量保存AJAX请求结果，每次操作本地变量，不用请求，减少请求次数  
   （3） 用innerHTML代替DOM操作，减少DOM操作次数，优化javascript性能。  
   （4） 当需要设置的样式很多时设置className而不是直接操作style。  
   （5） 少用全局变量、缓存DOM节点查找的结果。减少IO读取操作。  
   （6） 避免使用CSS Expression（css表达式)又称Dynamic properties(动态属性)。  
   （7） 图片预加载，将样式表放在顶部，将脚本放在底部  加上时间戳。
6. <https://github.com/Advanced-Frontend/Daily-Interview-Question>