1. 小程序sku

<https://www.cnblogs.com/kekeSummer/p/10103927.html>

1. 指定小程序某页面的二维码

<https://www.jianshu.com/p/558c4861191b>

1. 公众号要在同一个域名下执行
2. 小程序分享要用全局变量，而不是本地缓存，缓存ios获取不到
3. 缓存在组件设置缓存，在通讯中中无法第一时间获取，

解决方法：1、利用通讯返回的数据进行逻辑操作  
 2、利用全局变量进行逻辑操作

1. 定义组件时，要注意类型，Boolean时加双括号{{}} String不用
2. 小程序textarea问题：
3. line-height：
4. Ios高度过长，给定height
5. z-index过高，可以用条件判断，或者用它代替textarea

解决自定义一个组件

1. splice删除多个，splice会改变原有数组结构

因此对应下标也要随之改变

for(var i = 0; i < index.length; i++){

       cartNum -= cartsdata[idx].list[index[i]-i].count

       cartsdata[idx].list.splice(index[i]-i,1)

   }

1. 购物车数量要做缓存,购物车选中也可以缓存
2. 价格input封装组件（两位小数，最大、最小值），默认图片、标题等
3. Es6 new Set()  
    new Set() 对象存储的值总是唯一的  
    方法有：add、clear、delete、has、forEach  
    作用有：数组去重、并集、交集  
    参考：<https://www.cnblogs.com/ajaemp/p/11820339.html>
4. 各种正则<https://cloud.tencent.com/developer/article/1034345>  
    正则详细介绍 <http://caibaojian.com/javascript-zhengze.html>
5. 多个单值取最大最小：Math.min(a,b,c) Math.max(a,b,c)   
    数组取最大最小值 ：Math.min.apply(null,arr) Math.max.apply(null,arr)  
    相当于 Math.min(...arr) Math.max(...arr)
6. es6的详细介绍[https://es6.ruanyifeng.com/#docs/intro](https://es6.ruanyifeng.com/" \l "docs/intro)  
    <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Global_Objects/Object/defineProperty>  
    1、声明变量的6种方式 var function let const import class  
    1）var、function声明提前，破坏程序正常执行，内部变量可能覆 盖外部变量、可以重复命名、循环可能造成变量泄漏  
    2）let const块级作用域（绑定当前作用域），变量不可以提前， 否则造成暂时死区，不可以重复命名，const声明是必须同时赋值  
    3）import   
    2、解构 数组、对象、字符串、数值与布尔、函数参数、圆括号问题

左右类型要一致、默认值、对象键名要一致，新命名则{foo:bar}={foo:’12’} 、字符串伪数组  
 如果函数foo调用时没提供参数，变量x和y就不会生成，从而报错 通过提供函数参数的默认值，就可以避免这种情况。  
 function({x,y=5}) => 调用fun()报错

function({x,y=5}={}) => 正确,调用fun()不报错

3、函数的扩展  
 1) 默认值  
 2) rest参数后面不能再有参数，即为最后一个  
 function(a,...b) 正确   
 function(a,...b,c) 报错   
 rest参数为真正的参数，arguments为伪数组，需要通过 Array.prototype.slice.call(arguments)转为数组  
 3) es6 只要函数参数使用了默认值、解构赋值、或者扩展运算符， 那么函数内部就不能显式设定为严格模式，否则会报错,原因

参数先于函数体执行  
 解决1、全局性的严格模式 2、放在无参数的立即执行函数里面  
4) 箭头函数 固化this->继承下来的   
 以ES5表示ES6箭头函数，(1)函数外命名变量(2)bind(this)  
注意：以下不可以使用箭头函数

(1)对象里面的方法，不会得到预期结果

(2)动态改变this的时候

5) 函数尾调用 最后返回不做函数外的运算的函数  
 function(){return fn()}、

function(){if(a == 1){return fn1()}return fn2()}

错误  
6) 函数尾递归   
 递归非常耗费内存，因为需要同时保存成千上百个调用帧，很容易发生“栈溢出”错误（stack overflow）。但对于尾递归来说，由于只存在一个调用帧，所以永远不会发生“栈溢出”错误。  
 递归：函数自己调用本身，直到满足边界值结束(说白就是循环)

性能差

function factorial(n){

if(n === 1) return 1

return n \* factorial(n-1)

}

优化

function factorial(n,total=1){

if(n === 1) return total

return factorial(n-1,n\*total)

}

1. 构造函数不能使用 fun(){} 应使用 fun: function(){}

4、展开运算符  
 ES5 f.apply(null,[1,2]) ES6 f(...[1,2])

5、数组扩展  
 数组浅拷贝：var arr1 = [1,2]

var arr2 = arr1.concat()/arr1.slice() ES5

var arr2 = [...arr1] ES6

数组深拷贝

方法一：JSON.parse(JSON.stringfy(object))  
 局限性：会忽视undefined、symbol、函数

方法二：采用递归  
 function deepClone(obj){

let objClone = Array.isArray(obj)?[]:{};

if(obj && typeof obj==="object"){

for(key in obj){

if(obj.hasOwnProperty(key)){

//判断ojb子元素是否为对象，如果是，递归复制

if(obj[key]&&typeof obj[key] ==="object"){

objClone[key] = deepClone(obj[key]);

}else{

//如果不是，简单复制

objClone[key] = obj[key];

}

}

}

}

return objClone;

}

6、对象扩展  
 浅拷贝   
 Object.assign({},obj) 等于 obj1 = {...obj} 如果要复制的 值是一个取值函数，那么将求值后再复制。  
 拷贝值与原型   
 const obj2 = Object.assign  
 Object.create(Object.getPrototypeOf(obj)),obj  
 ）  
 深拷贝   
 1）JSON.parse(JSON.stringfy(object))

1. 优化性能 <https://www.cnblogs.com/mamimi/p/7646358.html>  
    <https://www.cnblogs.com/xiaohuochai/p/9178390.html>  
    1、方法、接口按需加载  
    2、抽离公共部分  
    3、图片懒加载  
    4、利用短路逻辑(注意return不能使用)、长度提前声明
2. 下拉刷新 this.onLoad() es6解构的没有属性会报错
3. <https://www.jb51.net/article/83938.htm>
4. 长列表 list:[[]] setData({[‘list[‘+(page-1)+’]’: res.data})  
    列表数据很多，列表很长，可能会有越往下拖动程序越慢的现象。有一次性set过多数据,解决分批set，即只set当前加载页
5. 采用async await方式时，要利用try{}catch(err){}以防止出错，采用 promise的then方式时，要利用then(res=>{}).catch(err){}以防止出错
6. 采用加密或者微擎框架发起请求时，在onLaunch中无法实例化，因此需 要用wx.request发起请求和url拼接siteinfo.js的配置
7. 请求列队
8. 调用组件时注意，组件内部的百分比是基于该组件的百分比，因此需要 直接表示组价百分比直接在组件加上class或者行内样式，此时组件样式和 内部样式直接受当前页样式影响，而不是组件延伸 的exterClasses的 class
9. 对象拷贝的方法 <https://www.cnblogs.com/hjson/p/10243806.html>
10. 封装混入mixins  
     //获取小程序
11. 时间 var time = +new Date() 等于 new Date().getTime()
12. 学习文档  
    <https://developer.mozilla.org/zh-CN/docs/Web/JavaScript/Reference/Operators/void>
13. wxs执行js慢
14. 登录状态全局化，一个账号放一个线上、一个体验的两套程序，同时授权session会过期，造成请求失败
15. 生成二维码要发布过的路径才能成功
16. 利用token在小程序正式版、体验版切换会导致重新登录，开发工具与正式版、体验版不能同时登录

三十一、理解扁平化

三十二、自定义navBar、tabBar时，page最好使用scroll-view嵌套内容，

按照渲染顺序，navBar、tabBar应该放在顶部，提前加载

三十三、聊天功能

<https://blog.csdn.net/qq_38912813/article/details/81540257>

三十四、控制七牛云图片大小

三十五、滚动区域没有开启惯性滚动 webkit-overflow-scrolling: touch  
三十六、节流，优化参考

<https://www.jb51.net/article/162093.htm>、 <https://www.cnblogs.com/plBlog/p/12053973.html>

<https://blog.csdn.net/baidu_25464429/article/details/90633841>

<https://zhuanlan.zhihu.com/p/147073485>

<https://blog.csdn.net/qq_16494241/article/details/90601124>  
<https://developers.weixin.qq.com/community/develop/article/doc/000008b94c41387644dadb61a58413>  
 <https://www.jianshu.com/p/d4fb22509eb9>

三十七、用fiddler、curl抓包检查前台的请求时间

<https://www.jianshu.com/p/5db6e1b173d5?from=groupmessage>

三十八、对象中的变量名等于变量值同名，可以利用结构

obj.name = name obj.price = price obj.title = title ==>

obj = {name,price,title}

验证对象是否为{} 有三种方式

1. JSON.stringify(obj) == ‘{}’
2. Object.getOwnPropertyNames(obj).length == 0
3. Object.keys(obj).length == 0

三十九、delect 对象特有，变量、原型是无法删除的，由于对象是引用地址（在对里面），delect会直接删除引用地址的变量名（即删除后无法再访问此变量）

四十、wx.createSelectorQuery()在真机上可能无法触发，可以利用setTimeout进行延迟执行（最后执行），利用骨架的时候尽量在骨架完成请求或者利用点击某个事件在获取其高度。对经常不变的库进行 Dll 打包，加快打包速度

四十一、设置配置代理需要重启编辑器

<https://www.jianshu.com/p/4c5c99abb864>

<https://www.imooc.com/wenda/detail/455752>

四十二、we-cropper插件 <https://github.com/we-plugin/we-cropper>

42、[https://icomoon.io/app/#/select](https://icomoon.io/app/" \l "/select) 合并icon

1. <http://caibaojian.com/h5-download.html> h5分享海报
2. 跨域问题 <https://segmentfault.com/a/1190000017312269>
3. \*\*创建封装、继承  
    封装 属性、方法私有化  
    实现方式: 闭包、\*构造函数、\*类  
    继承 子级可以访问到父级的方法、属性 通过原型继承的  
    实现方式: 定义原型方法、属性时 prototype.属性/方法
4. \*\*对象的克隆
5. 页面优化方案
6. 手写数组转树
7. 用setTimeout代替setInterval
8. 前端面试题第几条 1 11 17 18 30 39 42 74 89 99 117 128 135 143 147 149 158 163 169 181 189 190 191 192 199 207 228 232 237 246 254 261 262 269 171 273
9. 非行间样式 function getStyle(obj,attr){  
    return obj.currentStyle?obj.currentStyle[attr]:getComputedStyle(obj,false)[attr]  
    }
10. 理解offsetWidth scrollLeft
11. <https://www.jianshu.com/p/f1f39d5b2a2e> 面试题
12. scrollview 如何进行性能优化   
     <https://www.cnblogs.com/coco1s/p/5499469.html>  
     <https://zhuanlan.zhihu.com/p/30078937>
13. 深拷贝
14. <https://blog.csdn.net/yingzizizizizizzz?type=blog>  
     合成css sprite <http://tools.jb51.net/code/css_sprite>
15. Fixed在Ios和Android的解决方案  
     <https://blog.csdn.net/sinat_17775997/article/details/82690658>
16. Animation 详细介绍  
     <https://blog.csdn.net/u013243347/article/details/79976352>
17. 前端面试题<https://www.cnblogs.com/liandudu/p/14276948.html>  
     <https://blog.csdn.net/weixin_42003850/article/details/108651427>
18. Jquery 源码分析   
     <https://blog.csdn.net/qq_31967569/article/details/89556557>
19. 测试代码性能  
     <https://blog.csdn.net/dpj514/article/details/78767936>
20. HTML5+CSS3 最酷的 loading 效果收集  
    <https://www.runoob.com/w3cnote/free-html5-css3-loaders-preloaders.html>
21. Icon 引入问题  
     <https://blog.csdn.net/kingAn123/article/details/106459866/>  
     本地icon base64  
     <https://blog.csdn.net/liusunquan123/article/details/88885241>