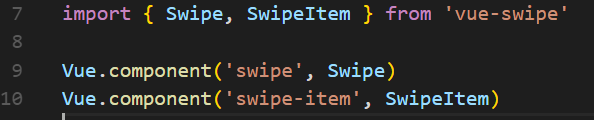
1. vue-swipe插件

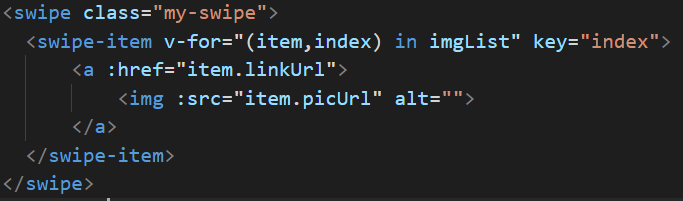
简单的slide效果

使用:

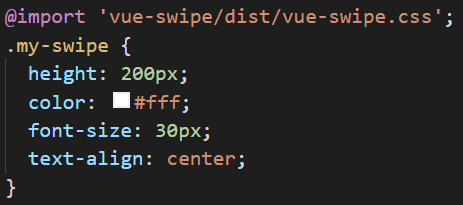
* $npm install vue-swipe –save
* 在main.js中引入



* 在需要的组件中使用

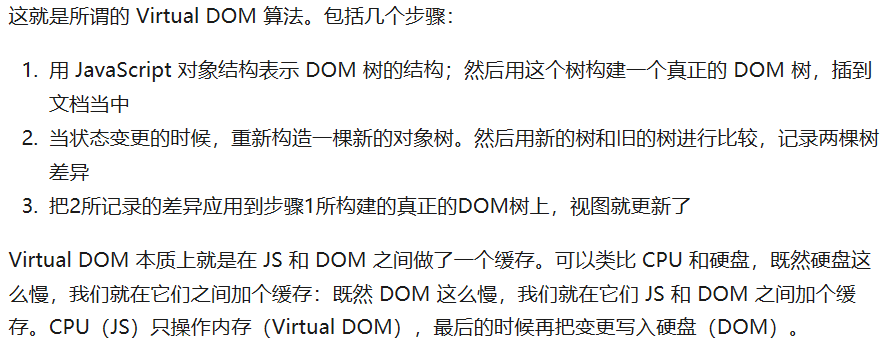


还要引入样式



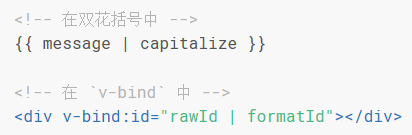
1. vue
2. 虚拟DOM

更新DOM的时间长开销大

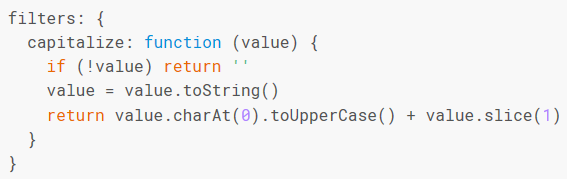


1. 过滤器

自定义过滤器：



定义本地过滤器：



定义全局过滤器



1. 开发环境/生产环境

开发环境下，Vue 会提供很多警告来帮你对付常见的错误与陷阱。而在生产环境下，这些警告语句却没有用

当使用 webpack 或 Browserify 类似的构建工具时，Vue 源码会根据 process.env.NODE\_ENV 决定是否启用生产环境模式，默认情况为开发环境模式。

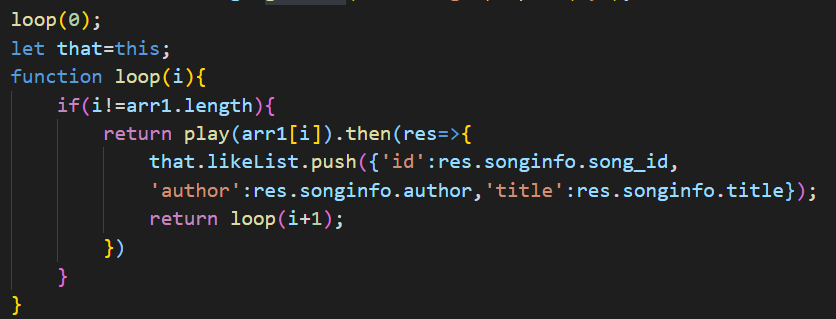


1. js
2. toFixed()会四舍五入 且转化为字符串



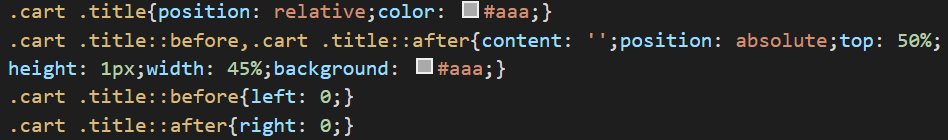
1. Promise遍历数组

递归：

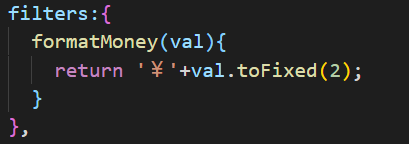


1. css
2. 文字在横线中间





1. vue
2. 对金额进行过滤，设置显示格式





1. 全局过滤器

Vue.filter(‘formatMoney’,(val,param1)=>{

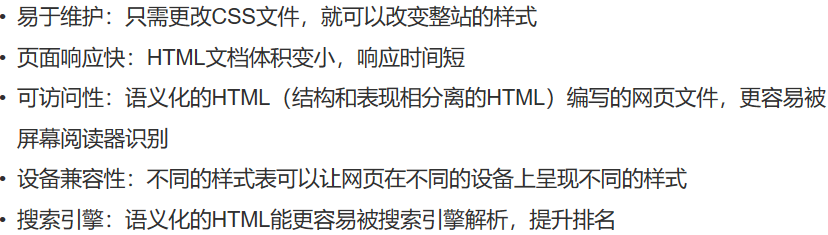
return “￥”+val.toFixed(2)+param1

})

{{price|formatMoney(param1)}}

1. 基础总结
2. 对web标准的理解

web标准是一系列标准的集合。网页主要由三部分组成：结构、表现和行为。对应的标准也分为3方面：结构化标准语言包括XHTML和HTML、表现标准语言主要包括CSS、行为标准语言主要包括ECMAScript等。这些标准大部分是由万维网联盟制定的，ECMAScript是ECMA制定的。web标准概括的讲就是结构、表现、行为三者分离。这样做的好处：



1. 浏览器内核

主要分成2部分：渲染引擎和js引擎。渲染引擎负责取得网页的内容、整理讯息、以及计算网页的显示方式，然后显示在屏幕上。js引擎解析和执行js来实现网页的动态效果。js引擎逐渐独立，内核主要指渲染引擎。

浏览器最核心的部分是”Rendering Engine”，即渲染引擎，一般叫做浏览器内核。它的功能是解释网页中的语法并渲染网页渲染引擎决定了浏览器如何显示网页内容以及页面格式信息。不同浏览器的内核不同，导致不同浏览器渲染出的页面可能有一些差异。

Trident内核：IE 360 搜狗 该内核几乎与W3C标准脱节，因为windows的市场占有率导致此内核使用比率也较大

Gecko内核 Netscape6及以上 代码完成公开

Presto内核 Opera7及以上 是一个动态内核，页面的全部或者部分都能够在回应脚本事件时等情况下被重新解析，执行js的速度最快

Webkit Safari Chrome 基于GPL开源

1. DOCTYPE

DOCTYPE位于html文档第一行，主要作用是为了告诉浏览器该以什么模式呈现。DOCTYPE不存在或不正确会导致文档以兼容(混杂)模式呈现。

通过document.compatMode这个属性可以检测页面的模式，CSS1Compat是标准模式，BackCompat是混杂模式。

标准模式的排版和js运作模式都是以浏览器支持的最高标准运行。在兼容模式中，页面以宽松的向后兼容的方式显示，模拟老浏览器的行为以防止站点无法显示。

H5为什么只用写DOCTYPE?h5不基于SGML(通用标记语言)，因此不需要对DTD进行引用，但是需要doctype来规范浏览器的行为。html4.01基于SGML。

标准模式和兼容模式的区别：

盒模型的解析 标准模式:border-box 兼容模式：content-box

当一个块元素(div)中包含的内容只有图片时，在标准模式下，所有浏览器在图片底部都有3px空白。在兼容模式下，Chrome中图片距div底部默认没有空白。

在混杂模式下，表格中的字体不会继承它的祖先元素

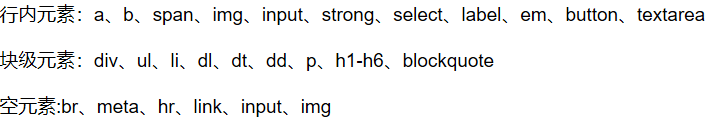
在IE的混杂模式下，给inline元素设置宽高都有效

在混杂模式下，IE浏览器对百分比宽度的解析不一样。如果父级是行内块或者浮动或者有定位的元素，给子元素设置宽度100%时，IE会以父级的100%算。标准浏览器取决于内容宽度。

在混杂模式下，给元素设置百分比高度时，其他浏览器正常(inline高度无变化，inline-block/block按百分比)，IE是自适应到内容高度。

overflow溢出默认值的问题。在标准模式下，溢出元素是可见的，超出部分的内容呈现在它的包含元素外。在兼容模式下，IE浏览器的溢出元素会自适应内容的尺寸。

1. 块级元素、行内元素、空元素



1. link和@important

link属于XHTML标签，@import是CSS提供的。

页面被加载时，link会被同时加载，@import引用的CSS会等到页面被加载完再加载。

import是CSS2.1提出的，ie5以上才能识别，link没有兼容性问题。

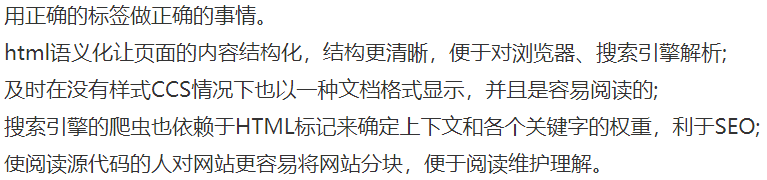
1. H5新特性

绘画canvas、媒介回放video/audio、本地离线存储localstorage长久存储/sessionStorage的数据在浏览器关闭时删除、语义化内容元素article footer header nav section、表单控件calendar date time email url search 新技术webworker websocket gelocation

移除的元素：纯表现的元素basefont/big/center/u/tt 对可用性产生负面影响的元素frame frameset noframes

IE8/7/6支持通过document.createElement方法产生的标签，可以利用这一特性让这些浏览器支持h5新标签，浏览器支持新标签后，还需要添加标签默认样式。

1. 对HTML语义化的理解



1. h5的离线存储

在用户没有与因特网连接时，可以正常访问站点或应用，在用户与因特网连接时，更新用户机器上的缓存文件。

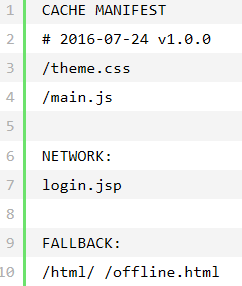
原理：HTML5的离线存储是基于一个新建的.appcache文件的缓存机制(不是存储技术)，通过这个文件上的解析清单离线存储资源，这些资源就会像cookie一样被存储了下来。之后当网络在处于离线状态下时，浏览器会通过被离线存储的数据进行页面展示。

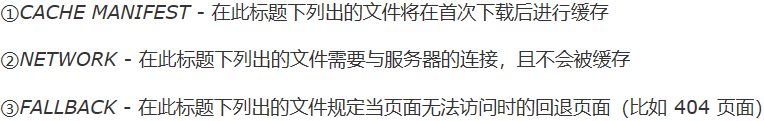
如何使用：

引入manifest文件



manifest文件的内容：





服务器上：manifest文件需要配置正确的MIME-type，即”text/cache-manifest”

浏览器如何对离线存储资源进行管理和加载：在线的情况下，浏览器发现html头部有manifest属性，它会请求manifest文件，如果是第一次访问app，那么浏览器就会根据manifest文件的内容下载相应的资源并且进行离线存储。如果已经访问过app并且资源已经离线存储了，那么浏览器就会使用离线的资源加载页面，然后浏览器会对比新的manifest文件与旧的manifest文件，如果文件没有发生改变，就不做任何操作，如果文件改变了，那么就会重新下载文件中的资源并进行离线存储。

1. cookie sessionStorage localStorage

cookie是网站为了标识用户身份而存储在客户端上的数据(通常通过加密)。cookie数据始终在同源htpp请求中携带(即使不需要)，即会在浏览器和服务器间来回传递。

sessionStorage和localStorage不会自动把数据发给服务器，仅在本地保存。

cookie数据大小不能超过4K。sessionStorage和localStorage比较大，5M左右。

localStorage存储持久数据，sessionStorage数据在浏览器窗口关闭时删除，cookie的数据可以设置有效期。

1. iframe的缺点

iframe会阻塞主页面的onload事件

搜索引擎的检索程序无法解读这种页面，不利用SEO

iframe和主页面共享连接池，而浏览器对不同域的连接有限制，所以会影响页面的并行加载。

所以，如果要使用iframe，最好使用js动态给iframe添加src属性值

1. label

label标签为input元素定义标记。

label标签不会向用户呈现任何特殊效果，为鼠标用户提高了可用性。

当用户选择该标签时，浏览器会自动将焦点转到和标签相关的控件上。

label通过for标签与控件相关联，for的值与id相同。

1. autocomplete

autocomplete属性规定浏览器是否启用自动完成功能，默认为on。当用户在字段开始键入时，浏览器基于之前键入过的值，应该显示出在字段中填写的选项。

1. 如何实现浏览器多个标签页之间的通信

WebSocket SharedWorker 或localStorage cookie等本地存储

1. 页面可见性

通过document.visibilityState可以检测当前页面是否可见以及打开网页的时间，visible/hidden。

应用：在页面被切换到其他后台进程的时候，自动暂停音乐盒视频的播放。

1. 如何在页面上实现一个圆形的可点击区域

map+area/svg

border-radius

纯js

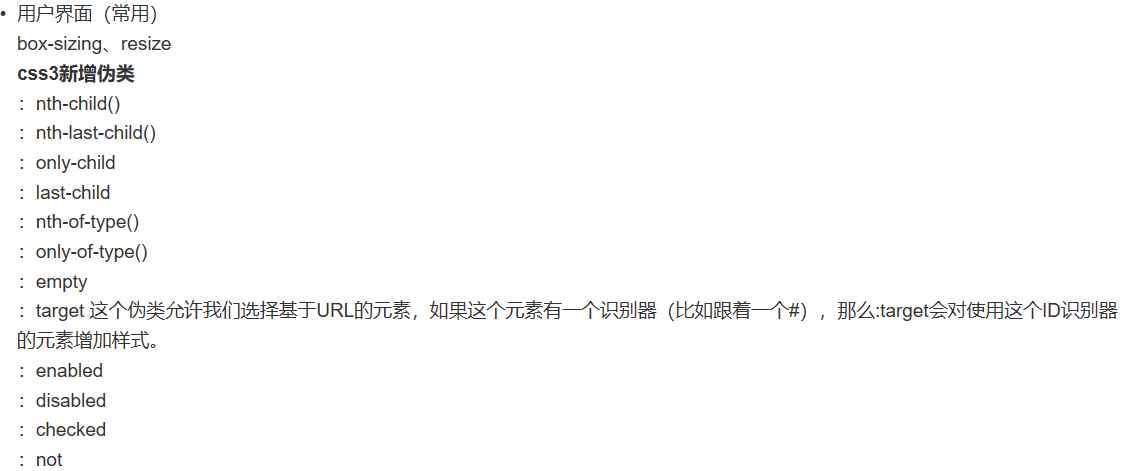
1. CSS选择器



1. CSS3新增属性

选择器、框模型、背景和边框、文本效果、2D/3D 转换、动画、多列布局、用户界面



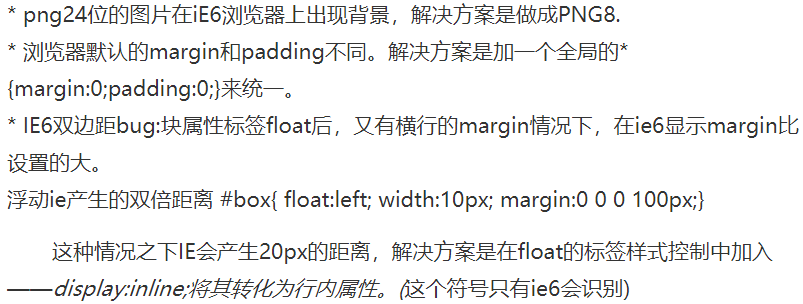


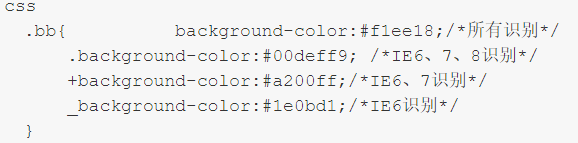
1. 纯CSS三角形

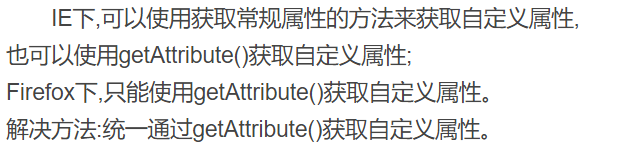


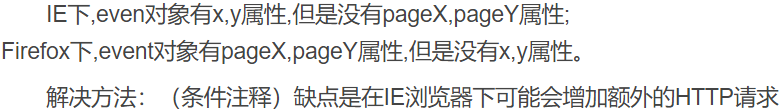


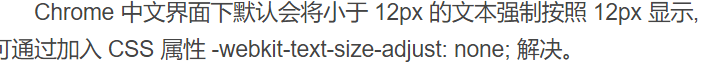
1. 常遇到的浏览器兼容性问题







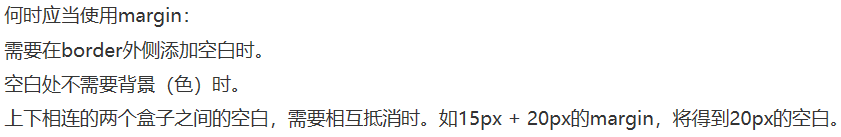


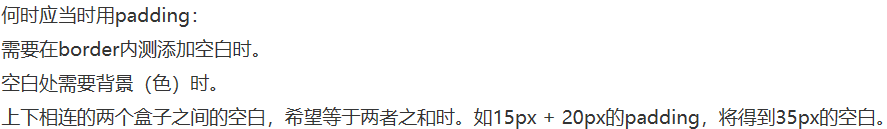


1. 为什么字体大小是偶数

习惯 在百分比布局时，换算单位和数值更精确方便

1. 什么时候用margin 什么时候用padding





1. font-style的italic和oblique

italic 斜体字 oblique倾斜的字体

1. 动画的最小间隔

多数显示器的默认频率是60HZ，即1s刷新60次，1/(60\*1000)=16.7ms

1. style写在body前后的区别

写在head中有利于浏览器逐步渲染。写在body标签后会导致浏览器停止之前的渲染，等待加载且解析完样式表之后重新渲染，在IE下可能会出现FOUC现象(样式失效导致的页面闪烁问题)

1. CSS预处理器和后处理器

预处理器例如：LESS、Sass、Stylus，用来预编译Sass或less，增强了css代码的复用性，还有层级、mixin、变量、循环、函数等，具有很方便的UI组件模块化开发能力，极大的提高工作效率。

后处理器例如：PostCSS，通常被视为在完成的样式表中根据CSS规范处理CSS，让其更有效；目前最常做的是给CSS属性添加浏览器私有前缀，实现跨浏览器兼容性的问题。

1. png jpg gif的区别



1. js异步编程的方法

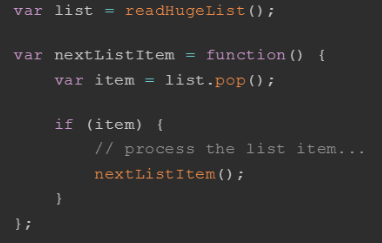
回调函数、事件监听、发布\订阅、Promise对象

1. str.match()字符串是否匹配这个正则表达式
2. setTimeout是window对象的方法，不是全局函数
3. typeof 无返回值的函数 结果是undefined
4. !的优先级比in高
5. 用立即执行函数封装功能模块的好处

创建一个私有空间，避免命名冲突

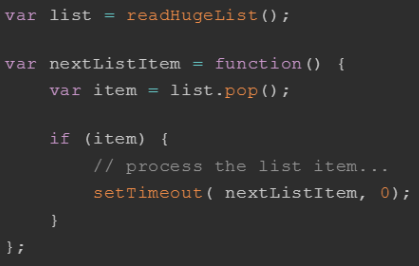
为全局变量提供一个易于引用的别名，如jquery的$

1. 堆栈溢出



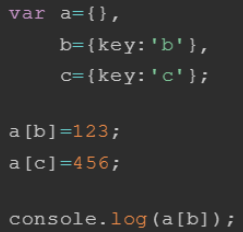
如果数组列表过大，上面的代码会导致堆栈溢出，如何解决这个问题并保留递归？

注：递归指函数调用自身

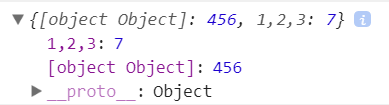


堆栈溢出被消除，因为事件循环处理递归，而不是堆栈。

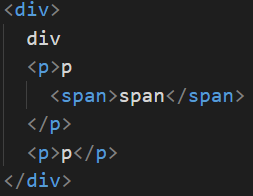
1. 设置对象属性



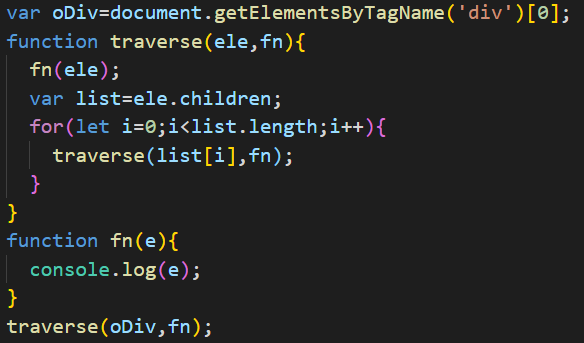
输出的结果是456 因为设置对象属性时，js会隐式的将属性名字符串化



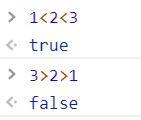
1. 深度优先遍历和广度优先遍历



深度优先遍历：



1. js引擎对<>的关联性的工作



3>2>1被翻译为true>1

1. 如何在数组的开头/末尾添加一个元素

arr.push末尾 arr.unshift开头

还可以借助es6的扩展运算符

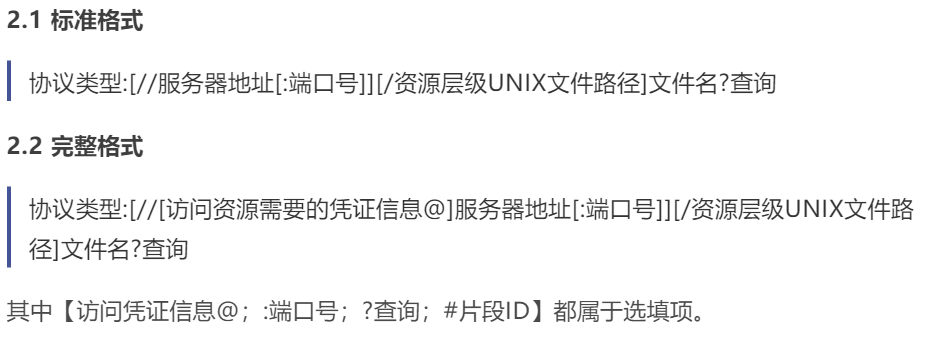


1. url

统一资源定位符 俗称网址

Uniform Resource Locator

url格式：



语法规则：



scheme 因特网服务类型，即协议类型，最常见的是http

host 定义域主机 http默认主机是www

domain 定义因特网域名

port 端口号 http默认端口号是80

path 定义服务器上的路径

filename 定义文档/资源名称

主要的scheme:



url类型：绝对url 相对url

.代表当前目录 ..代表上层目录 ../../代表上层的上层 /代表根目录，绝对路径

href指定网络资源的位置，目的不是为了引用资源，而是建立联系，让当前标签能够链接到目标地址

src指向外部资源的位置 指向的内容会将应用到标签当前所在位置

href和src的区别：

请求资源类型不同：href指向网络资源的位置，只建立联系；src会下载资源并应用到文档中

作用结果不同：href用于建立联系，src用于替换当前内容

浏览器解析方式不同：遇到href，浏览器会并行下载资源而且不会停止对当前文档的处理，这也是建议使用link不用@import的原因；遇到src，浏览器会暂停其他资源的下载和处理，这也是建议把js放底部而不是头部的原因

link和@import的区别：

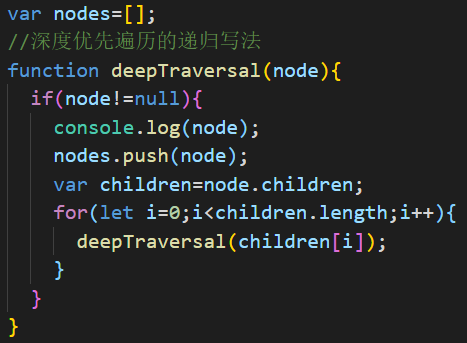
link是XHTML标签，除了能够加载CSS，还可以定义RSS等其他事务；@import属于CSS，只能加载CSS

link引用CSS时，在页面载入的同时加载；@import需要等页面完全载入后再加载

link是XHTML标签，无兼容性问题；@import是CSS2.1提出的，低版本浏览器不兼容

link支持使用js控制DOM改变样式，@import不支持

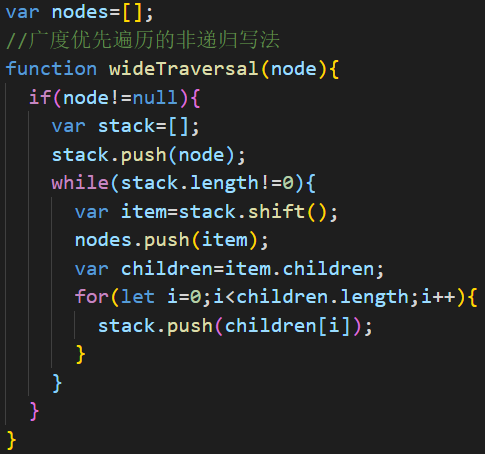
1. 深/广度优先遍历



如果不用递归，需要借助栈的后进先出特性，这样导致右子树先遍历



广度优先遍历要借助队列的先进先出特性

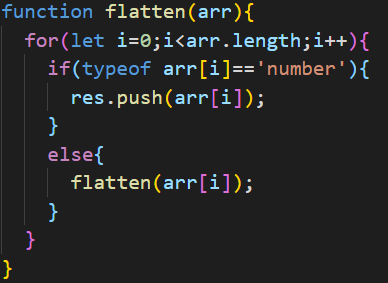


1. 数组扁平化

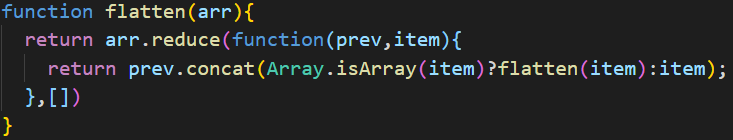
方法1：扩展运算符 只适合二维数组

方法2：toString() 先转换为字符串，再用split(‘,’)分隔成数组

方法3：递归



方法4：reduce



注：递归指函数调用自身，回调指函数做参数并在函数末被调用

1. 框架和库的区别

框架要求你用某种形式写代码，有自己的语法特点，对应各个模块

库专注于一点，提供了写代码时某种形式的写法

框架的好处：

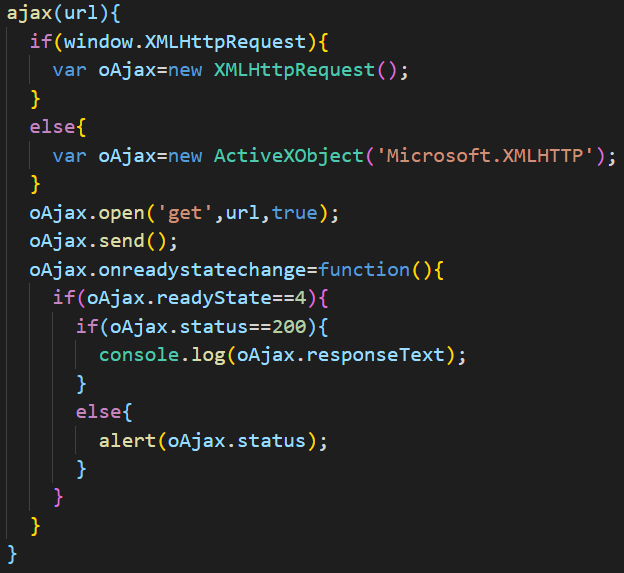
提高代码质量和开发速度

提高代码的复用率

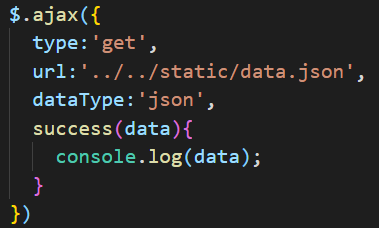
降低模块间的耦合度

1. 获取json文件里的数据(json文件要放在static文件夹里)

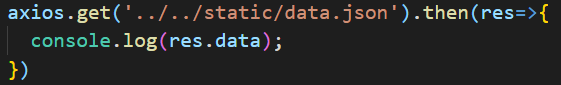
使用原生ajax



使用jquery的$.ajax()



使用axios:

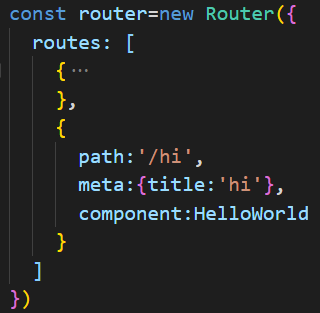


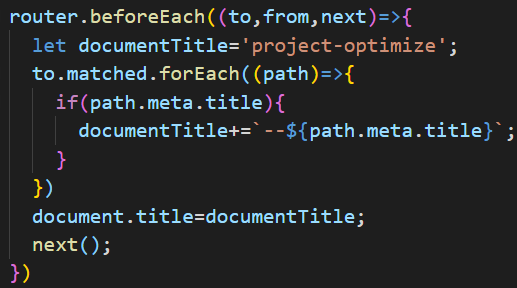
注：引入jquery:$npm install --save jquery

引入axios:$npm install --save axios

1. 动态修改document.title

在路由元信息meta里设置title,然后在router.beforeEach钩子函数里修改





1. event bus 主要是兄弟组件通信

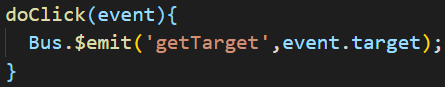
点击Click组件的按钮，触发Show组件的事件

首先，新建一个空的vue实例

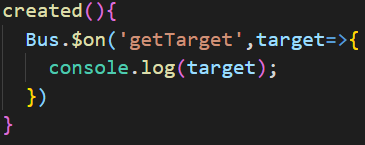


然后，在Click组件里设置点击事件，触发Show组件的事件





最后，在Show组件中设置要被触发的事件

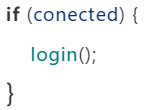


注：在Click组件和Show组件要引入Bus

1. easy mock做模拟数据接口的网站
2. 使用!!操作符转换布尔值

如果使用!!variable检测变量的值，只要变量的值是：0 null ‘’ undefined NaN都返回false

1. 使用+将字符串转换成数字，不能转换则返回NaN，如果是Date对象则返回时间戳
2. 并条件符



可以用&&简写



1. 在循环中缓存arr.length

如果数组比较大，每次循环都要计算数组长度所以可以把数组长度赋值给一个变量

1. 检测对象中的属性

in或obj.hasOwnProperty()

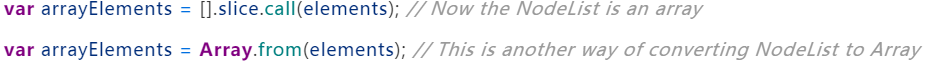
1. 合并数组

一般会使用concat，但数组较大时会消耗大量内层来存储新创建的数组，所以可以用Array.push.apply()

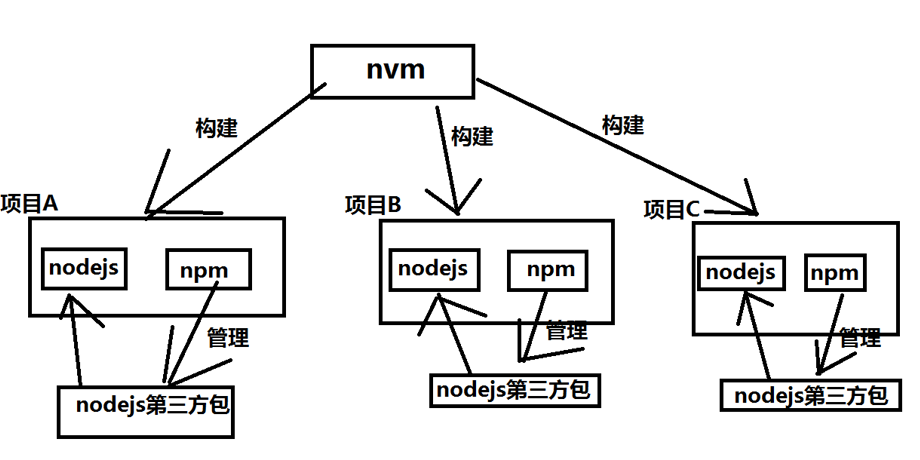


注：concat不改变原来的数组

1. 将nodeList转换成数组



1. nvm管理node版本，npm是node的包管理工具



1. 构建工具

grunt gulp npm-scripts webpack fis prepack rollup

推荐：webpack+npm-scripts

使用构建工具的好处：资源压缩、静态资源替换、模块化处理、编译处理

1. 安装sass

$cnpm install --save-dev node-sass 推荐使用cnpm

1. 修改eslint规则

修改.eslintrc.js文件中的rules



详细信息：https://segmentfault.com/a/1190000008742240

1. vue-router入口路径重定向

入口页面为商品页面



1. vue是MVVM框架

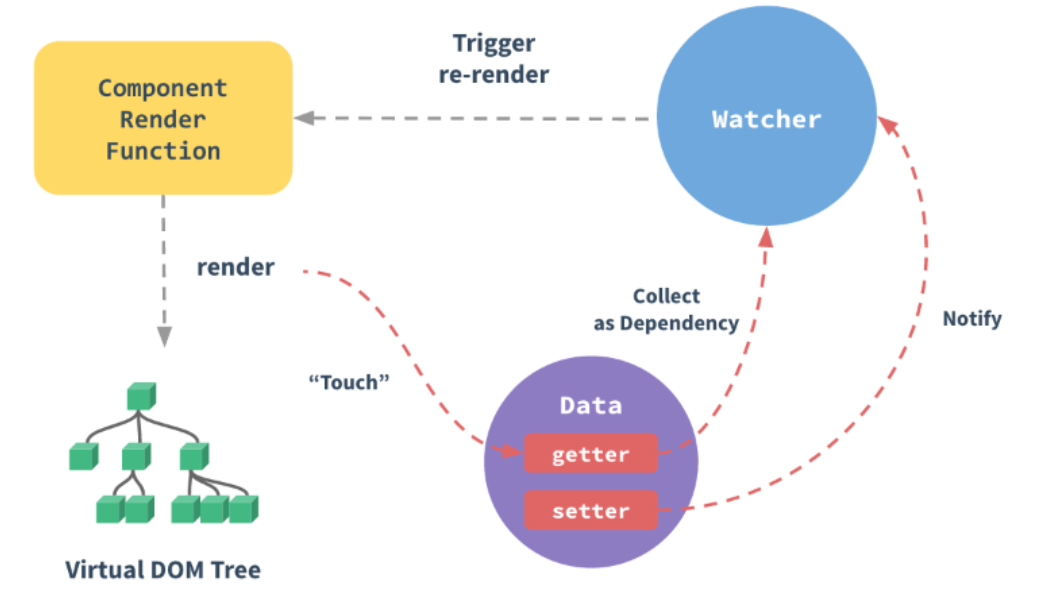
vue是轻量级框架，核心思想是数据驱动和组件化

MVVM框架：

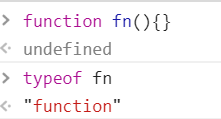
view <-----> ViewModel <----->Model

视图(DOM) 中间件 数据(JS)

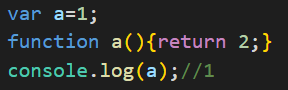
数据改变驱动视图自动更新：



1. 使用typeof可以检测出是不是函数



1. 函数声明会覆盖变量声明，但不会覆盖变量赋值



1. var f=function foo(){}中foo只在函数内部有效
2. sass
3. 使用变量

sass用$来标识变量

声明sass: $highlight-color

使用sass: $highlight-color: #F90;

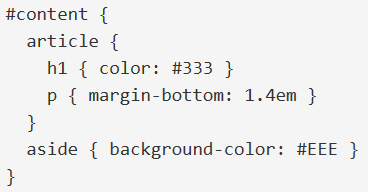
变量在规则块外面定义的为全局变量，在规则块内定义的只能在当前规则块内使用

变量值也可以引用其他变量

凡是css标准值可存在的地方，变量就可以使用

用中划线声明的变量可以使用下划线的方式引用，反之亦可

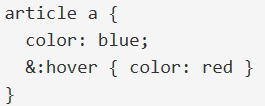
1. 嵌套规则



群组选择器，子选择器，弟选择器，同级选择器(~)都可以嵌套

&代表父类

伪元素：



相当于



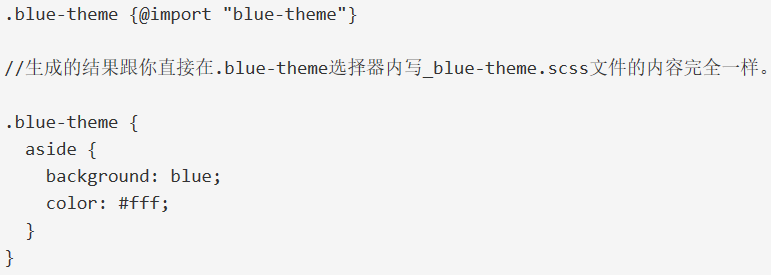
1. 导入sass文件

@import导入 且在生成css文件时就把相关文件导入进来

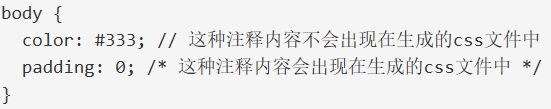
那些专门为@import命令而编写的sass文件，并不需要生成对应的独立css文件，这样的sass文件称为局部文件。sass局部文件的文件名以下划线开头

!default用于变量，含义是：如果这个变量被声明赋值了，那就用它声明的值，否则就用这个默认值

嵌套导入：



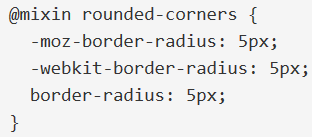
静默注释：内容不会出现在生成的CSS文件中

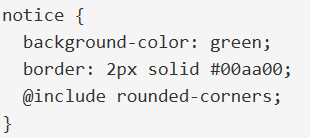


事实上，如果注释出现在原生css不允许的地方，这些注释也会被抹掉

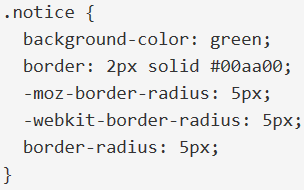
1. 混合器

@mixin定义混合器 @include使用混合器



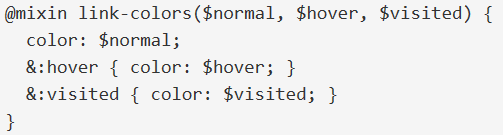


最终生成的是：



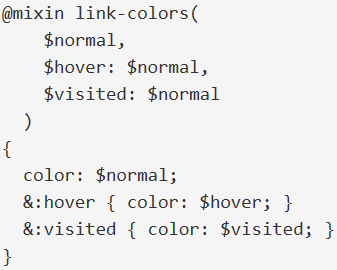
混合器中不仅可以包含属性，还可以包含css规则，还可以用&

给混合器传参：像函数参数一样



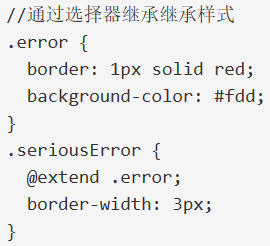


参数默认值：



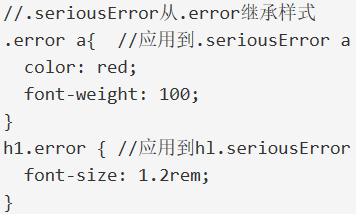


1. 继承样式 @extend



元素最终的展示效果就好像是class="seriousError error"

.seriousError不仅会继承.error自身的所有样式，任何跟.error有关的组合选择器样式也会被.seriousError以组合选择器的形式继承



1. vscode中vue文件不高亮

安装vetur插件

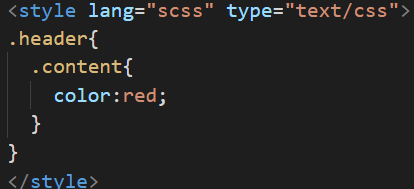
1. vscode使用sass

cnpm install node-sass --save-dev

cnpm install sass-loader --save-dev

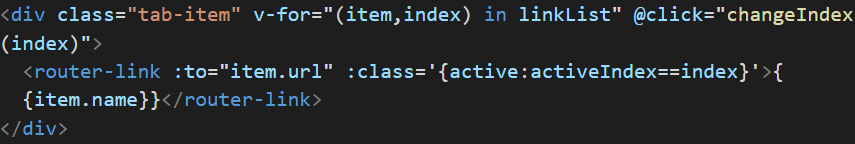
在webpack.base.config.js中的rules数组中添加：

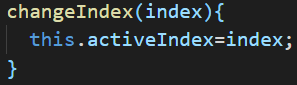




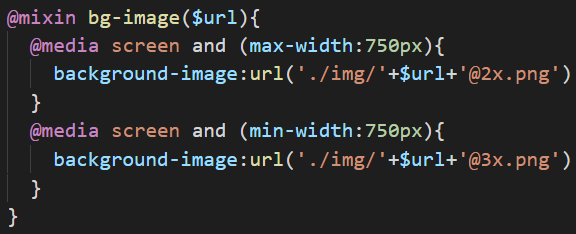
1. 路径切换

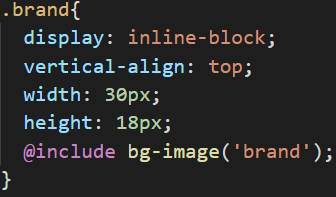
声明一个变量activeIndex:0



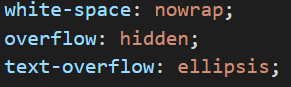


1. 使用sass的混入实现自适应布局



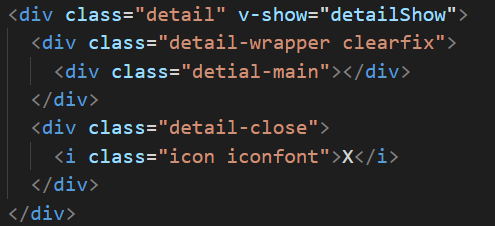


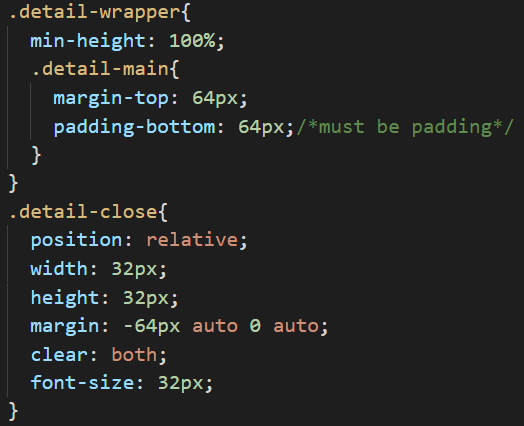
1. 超出部分用…



1. sticky footer

如果页面内容不够长，页脚粘在底部；如果内容足够长，页脚被内容向下推送，不会覆盖内容



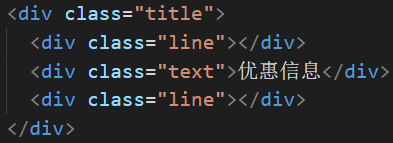


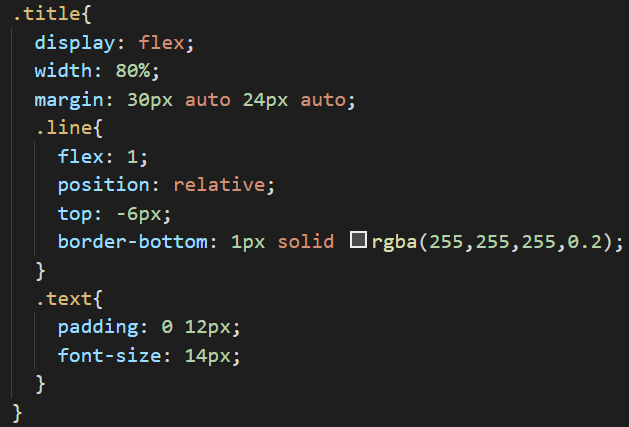
1. 向下取0.5倍数

let score=Math.floor(this.score\*2)/2;

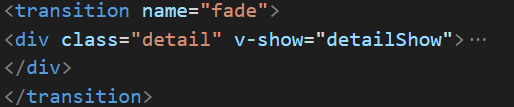
1. 标题 两边有横线



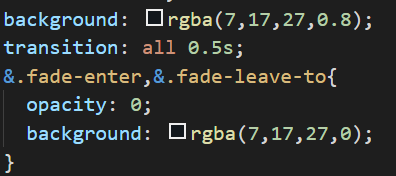




1. 渐入渐出效果



.detail：



1. better-scroll插件

内容滚动 内容始终不超过父容器

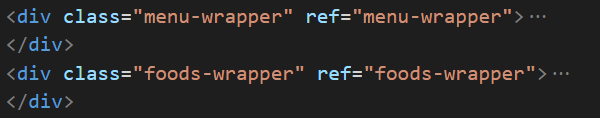
* 1. 安装

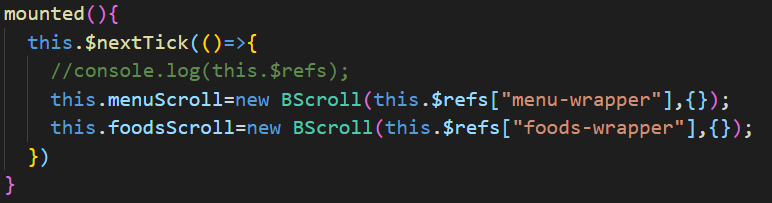
$ cnpm install better-scroll

* 1. 引入

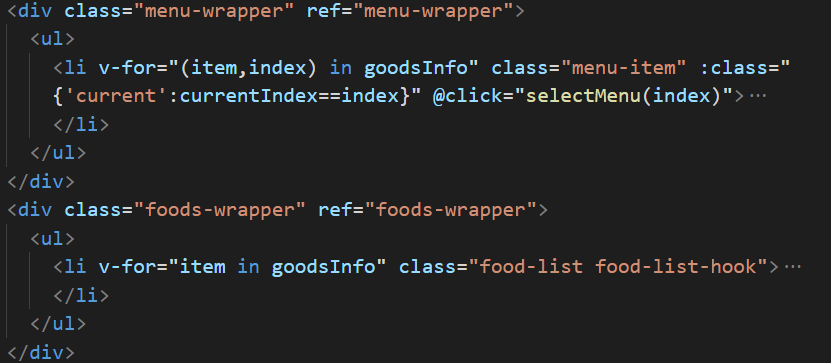
import BScroll from 'better-scroll'

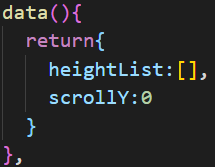
* 1. 使用



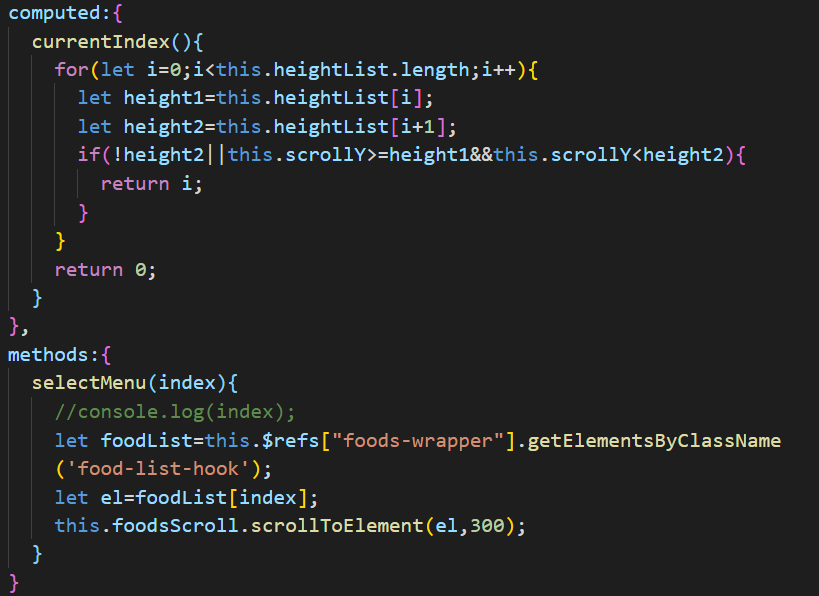


高亮效果：

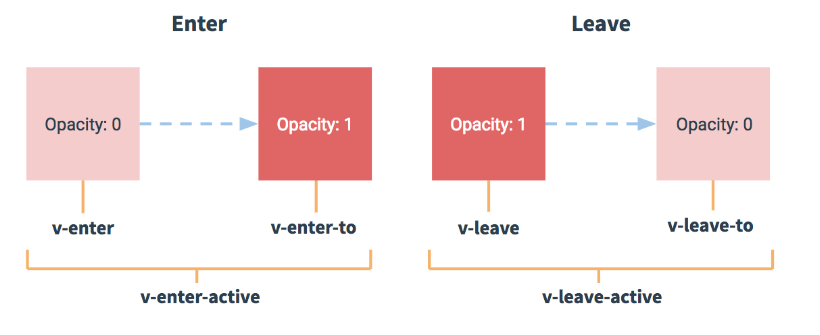


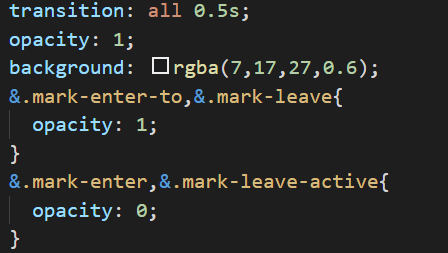






1. transition





1. 阻止冒泡和默认事件



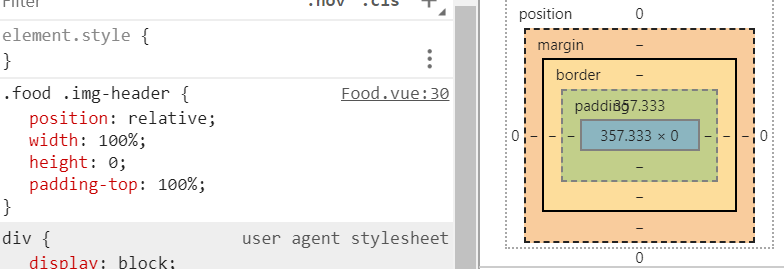
这样，点击子元素不会触发父元素的点击事件

1. 用ref实现父组件触发子组件的事件

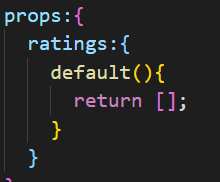




1. 图片宽高和浏览器宽度一样



1. 设置props的默认值

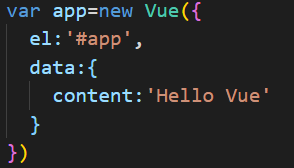


还可以设置类型 type:Array

1. v-show也可以绑定一个函数



1. Vue不支持ie8及以下浏览器
2. 实例和配置项



app为创建的vue实例

el和data等为配置项

el指定vue实例要接管的dom节点

1. live-server插件

保存更改后自动刷新

要右键 open with live server

1. $.proxy(fn,content)

为函数指定特定的上下文

1. mvp模式

m数据 v视图 p控制器

视图层发生变化通知控制器，控制器操作dom

view model 不直接联系

大量dom操作

1. 生命周期函数

beforeCreated

created 初始化

beforeMount 模板和数据结合 未挂载到页面上之前

mounted 挂载到页面上后

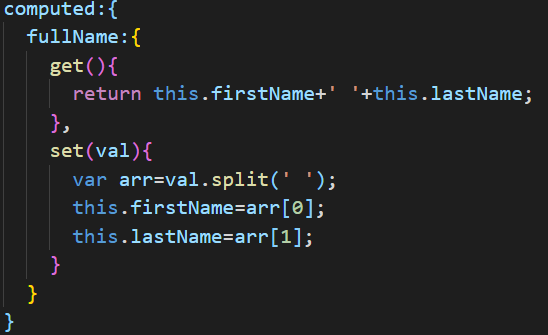
beforeUpdate updated 当数据发生改变前后

beforeDestroy destroyed 被销毁(调用vm.$destroy())前后执行

1. 计算属性computed和watch都内置缓存

计算属性和watch依赖的变量没有发生变化时，不会再重新计算

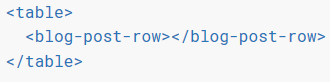
1. computed不仅可以获取还可以设置



1. template模板占位符

template 包裹一些元素 用v-for循环 结果中template不会表现出来

1. li tr等只能出现在某些特定元素的内部



上面的写法不对 要用is属性



如果从以下来源使用模板，这个限制不存在：



1. data应该是一个函数

如果是一个对象，不同组件的数据会相同

函数可以返回不同的对象

1. ref 给元素或子组件注册引用信息

方便引用

指向DOM元素 或子组件实例

通过vm.$refs.name 获取

ref是作为渲染结果创建的，在初始渲染时不存在还不能引用

1. 子组件不能修改父组件传递的值，可以复制给一个新变量再修改，也不会影响父组件中的值
2. 非props特性

子组件没有接收的属性

会显示到DOM上(添加到这个子组件的根元素上)

1. 绑定原生事件

子组件通过emit触发父组件的事件

绑定原生事件：加native修饰符 不用在定义的事件中用emit触发父组件的事件 直接使用父组件的原生事件

1. 非父子组件间传值

Event Bus 总线 发布订阅模式 观察者模式

1. 插槽slot

优雅的插入DOM 向子组件传递DOM

父组件传递给子组件的内容把slot代替 不会有无用的标签

slot标签中间可以设置默认值

slot只能有一个没有name属性

作用域插槽：可复用的插槽 可以循环列表 slot-scope=”item” 要用template标签(2.5.0+不限制只在template上)

eg.

child子组件的模板：

`<div> <p>hi</p> <slot></slot> </div>`

父组件中：

<child> <p>vue</p> </child>

结果：

<div> <p>hi</p> <p>vue</p> </div>

1. 动态组件

根据is属性的变化加载不同的组件

在动态组件上可以使用keep-alive缓存 避免重复加载

1. v-once 只渲染元素和组件一次 包括子节点
2. transtion的name属性默认为v

如果不设置name 则类名为v-enter等

1. animate.css库在vue中的使用

引入

< v-enter-class=”…”>

1. transition和animate.css配合

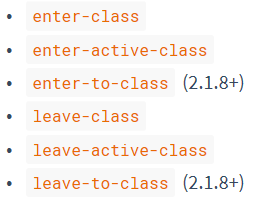
引入animate.css

<transition>

<div enter-active-class=’animated bounce’></div>

</transition>

自定义过渡类名：



1. 显性的设置动画过渡时间

如果动画和过渡一起使用，以谁的时长为准？





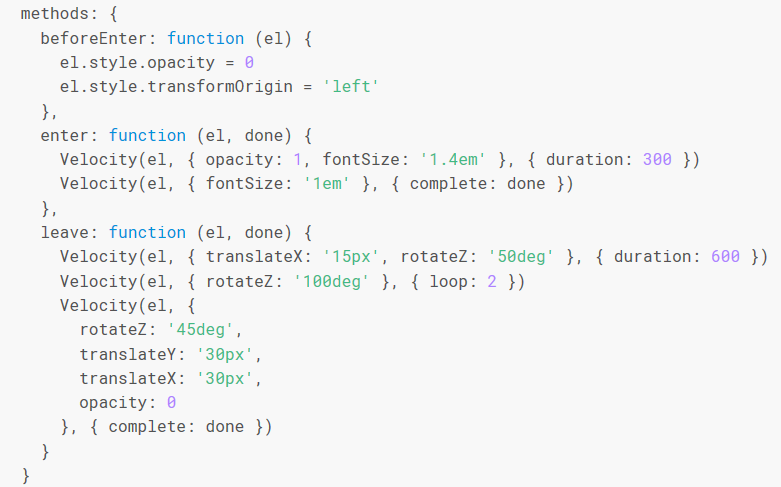
1. transition钩子



接收参数el 即发生动画的标签



使用Velocity.js



1. 多元素/多组件过渡

多元素借助key属性

多组件使用动态组件

1. 移动端点击事件有300ms延迟

双击缩放，所以浏览器在第一次单击之后会延迟300ms再触发点击事件，等待这段时间内时候有第二次单击，如果有浏览器会按双击处理。

解决办法：

禁止缩放

fastclick.js

vue中如何使用fastclick.js

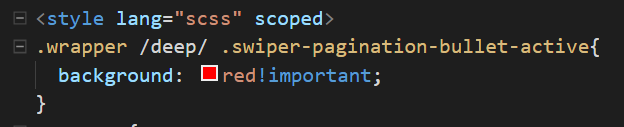
main.js中





1. 样式穿透

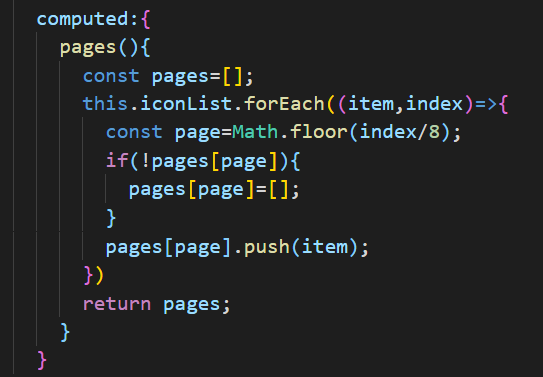
scoped的样式只能用于当前文件，如果使用了第三方插件就不好改变插件的样式

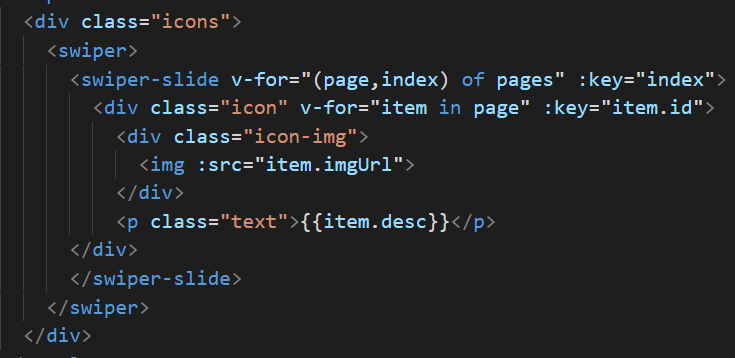


sass用 /deep/

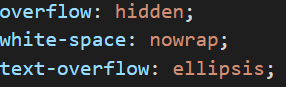
stylus用>>>

1. 多页轮播

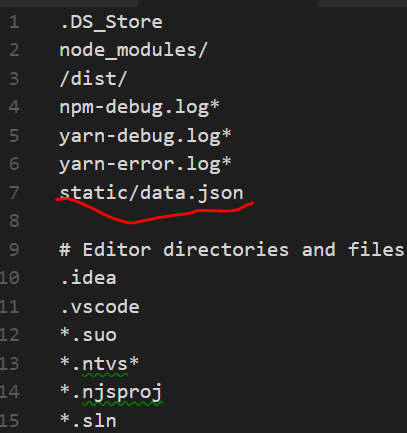




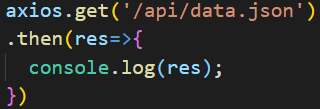
1. 单行文本超出省略



1. .gitignore文件里可以声明一些不会被提交到远程仓库的文件

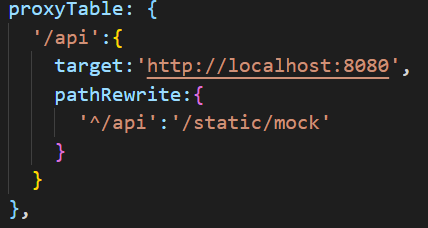


1. proxyTable 接口代理



axios访问的是api下的数据，实际访问的是static/mock.data.json

config/index.js中设置：



1. better-scroll的结构要求

div

div-wrapper

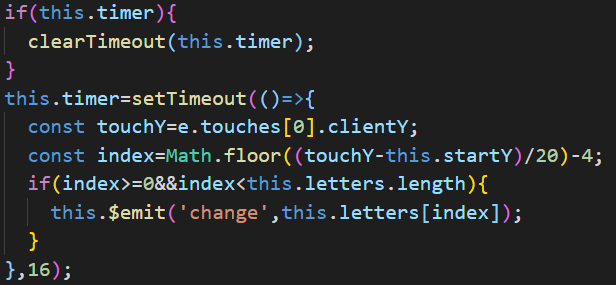
item

item

1. 函数节流

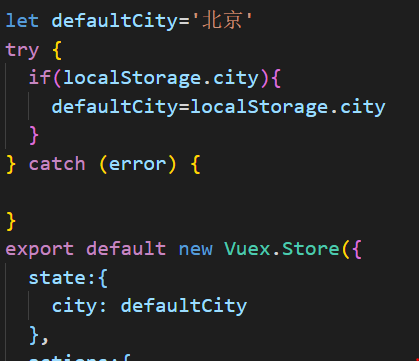
鼠标的mousemove scroll等事件都是在短时间内重复触发的。

DOM操作比非DOM交互需要更多的内存和CPU时间，连续尝试进行过多的DOM相关操作可能会导致浏览器挂起，有时候甚至会崩溃。为了解决这个问题，需要使用定时器对该函数进行节流。



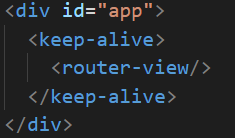
1. 使用localStorage建议使用try-catch包裹

如果用户使用隐身模式或关闭本地存储功能，使浏览器抛出异常，项目不能运行。



1. vue每次切换页面都重新请求ajax

解决办法：keep-alive标签



把ajax存放在内存中，第二次请求页面及以后，不再发生ajax请求，而是从内存中获取

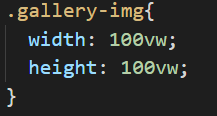
1. router-link

默认渲染成a标签

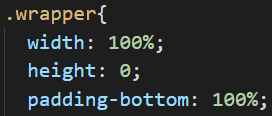
如果不想渲染成a标签：



1. 令图片宽100%，高和宽相等



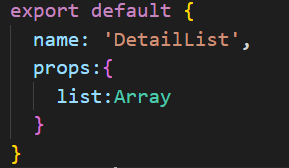
1. 让图片未加载前仍占据空间，图片后面的元素不会在图片加载后下移

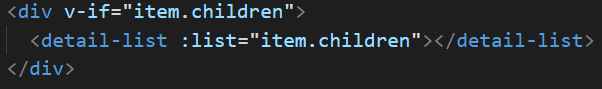


图片用div包裹一下

1. 递归组件

组件自身调用自己



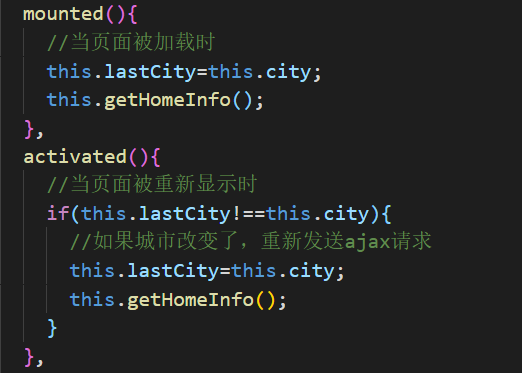


name的作用之一就是在递归组件中调用自己

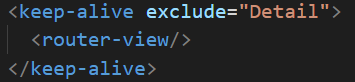
1. keep-alive缓存包含的所有

如果想重新发送ajax请求，方法：

方法1：activated生命周期钩子



方法2：exclude

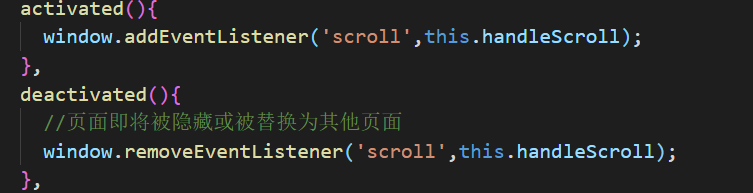


必须给Detail组件设置name属性



1. 解除事件绑定

如果在某组件给window加事件绑定，那么在其他组件这个事件需要解除绑定



1. vue实例中name的作用

递归组件中引用自身

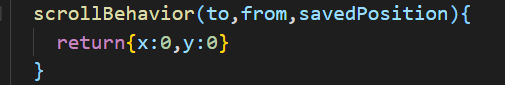
exclude中使用

vue-devtools中组件名字

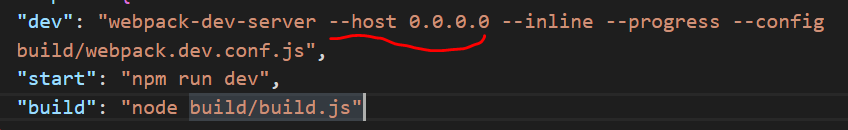
1. 在组件1中滚动到底部再打开组件2，组件2显示的是下面的内容

解决办法：

在router/index.js的router实例中添加：



1. vue项目在手机上运行





在main.js中引入：



1. 异步组件

vue项目在初始的时候app.js请求了所有页面的代码

代码过多的时候，app.js请求的速度就慢，所以应该打开某页面时再请求某页面代码



把component的import方式改成import

注：app.js比较大，几MB时才用异步组件

1. 什么是小程序

小程序是一种吧需要下载安装即可使用的应用。它实现了应用“触手可及”的梦想，用户扫一扫或者搜一下即可打开应用。也体现了用完即走的理念，用户不用关心是否安装太多应用的问题。应用将无处不在，随时可用，但又无需安装卸载。

1. 小程序能做什么

微信小程序是一种全新的连接用户与服务的方式，它可以在微信内被便捷的获取和传播，同时具有同原生APP一样出色的使用体验。

通俗的说，微信小程序可以干90%原生APP能干的事情，小程序无需安装、易传播，但它又不是万能的。

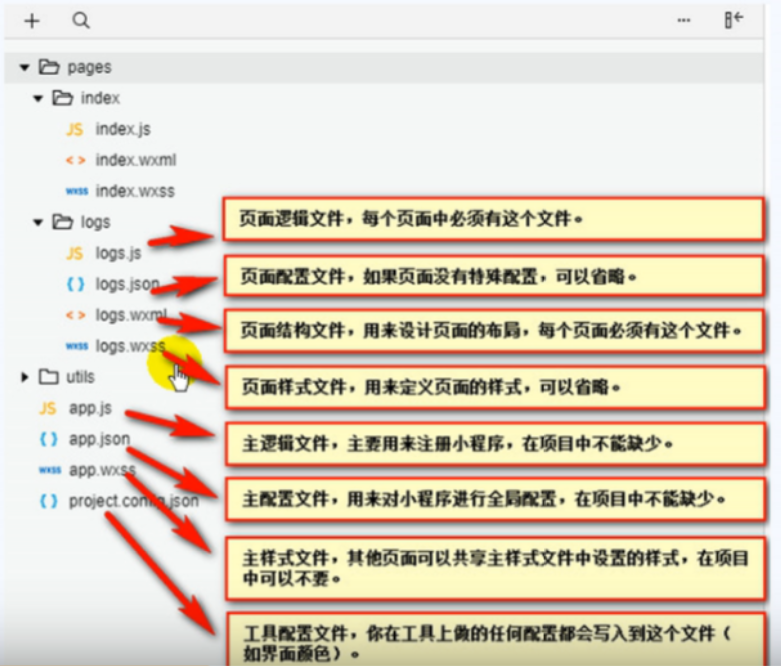
1. 小程序和原生APP的区别



1. 小程序的优点

无需安装 、用完即走、触手可及、无需注册、无需登陆、社交裂变

1. 小程序的目录结构



1. 为了方便开发者减少配置项，每个页面的4给文件要有同样的路径和文件名
2. 页面路径配置

主配置文件app.json中pages配置项指定小程序由哪些页面组成，数组第一项代表小程序的初始页面。

文件名不需要写后缀，小程序会自动寻找路径下.json .js .wxml .wxss4给文件进行整合

1. page的初始化顺序



先加载，然后显示，然后渲染完成

1. 下拉刷新

使用onPullDownRefresh()必须在配置文件中将enablePullDownRefresh设置为true

1. setData()

同步修改this.data，异步修改视图层

1. 小程序的特点

适合做简单的、用完即走的应用

适合低频的应用

适合性能要求不高的应用

1. 适合小程序的应用

饿了么、猫眼电影票、滴滴打车

不适合：知乎、斗鱼直播、阴阳师

1. text

text标签 和不使用 显示是一样的

但是只有用text标签包括的文字才能才按选中

1. style和class

style里应该设置动态样式，静态样式写样式表里(根据class)

1. \n

<text>Hel \n lo</text>

会换行显示

1. 分辨率和rpx

rpx可以根据屏幕宽度自适应。规定屏幕宽度为750rpx。

如iphone6，屏幕宽度375px,共有750个物理像素，则750rpx=375px=750个物理像素，1rpx=0.5px=1物理像素。设计师的图一般是750px。

1pt包含1/2/3个px

pt为逻辑分辨率，pt大小和屏幕尺寸有关，可以理解为长度和视觉单位。

px指物理分辨率，px是像素点

1. “”布尔值

小程序把””里面的当作字符串

vertical=”false” 结果是垂直的

vertical=”{{false}}” 结果是横向的

1. setData





1. navigateTo与redirectTo

redirectTo 2个页面平级，没有返回键 页面onUnload

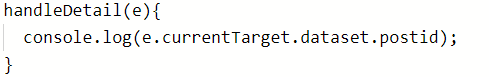
navigateTo 从父页面到子页面，子页面有返回键 最多有5级 页面onHide

1. bindtap catchtap

catchtap阻止冒泡

1. require脚本文件只能用相对路径
2. 自定义数据





大写会自动转换为小写，连字符会转换为驼峰

data-postId -> dataset.postid data-post-id -> dataset.postId