1. svg代码转换地址：<https://www.zhangxinxu.com/sp/svgo/>
2. git push origin dev:refs/for/dev

【**工作日常笔记：**】

1.git使用

(工作日常用)

(第一次用的话，设置密钥，你百度一下，或者让同事帮忙搞一下，很简单的)

//查看本地分支

git branch

//查看远程分支

git branch -r

//切换到分支:

git checkout (分支名)

eg:切换到分支:lanjian

git checkout lanjian

git add .

git commit -m "描述一下（eg:项目完成了登录注册功能）"

git push origin 分支名:refs/for/分支名

eg:(push到远程的lamjian分支)

git push origin lanjian:refs/for/lanjian

git push origin HEAD:refs/for/lanjian

（// 将本地分支与远程同名分支相关联

git push --set-upstream origin <本地分支名>）

（以前整理的）

1. git - 简易指南：

（1）http://www.bootcss.com/p/git-guide/

（2）<https://blog.csdn.net/Hanani_Jia/article/details/77950594（参考）>

（3）https://www.liaoxuefeng.com/wiki/896043488029600/900003767775424（廖雪峰）

（4）https://learngitbranching.js.org/

    git config --global user.name "xxl-007"

    git config --global user.email "1186479152@qq.com"

    git clone 地址 //克隆对应地址的远程仓库

    cd 文件夹名 //进入哪个文件夹

    git add . //添加到本地仓库 .代表所有，也可以写对应的文件

    git status //查看状态

    git commit -m "" //引号内填写本次提交的备注

    git push origin master //提交到远程仓库的master分支

    git pull origin master： 拉取远程代码，实现同步

    git init：   创建本地创库

    git remote add origin +远程创库地址：         本地创库与远程仓库连接

    git add +文件名称：  提交代码到文件夹里，真正把文件提交到仓库，

    git commit -m "": 进行备注解释

    git pusn origin 分支名称:提交到远程仓库。第一次提交需要加-u，设置默认分支。

    git status:查看状态

    git diff +文件名称: 查看提交文件修改内容

    git log --pretty=online :查看历史

    git reset --hard HEAD^ : 历史回退

    git reflog:查看所有操作记录

    git reset --hard 版本号：根据操作记录回退

    git branch 分支名称：创建分支

    git checkout 分支名称：切换分支、、、-b即创建又切换分支

    mkdir: 创建文件夹

    touch:创建文件

    pwd：查看当前文件路径

    ls:查看当前文件夹所有的东西

    cat 文件名称： 查看文件内容

    ssh-keygen -t rsa -C <邮箱名称>：获取密钥

    cat ~ssh/id\_rsa.pub:查看密钥

git merge 分支名称：合并分支

要关联一个远程库，使用命令git remote add origin git@server-name:path/repo-name.git；

关联后，使用命令git push -u origin master第一次推送master分支的所有内容；

此后，每次本地提交后，只要有必要，就可以使用命令git push origin master推送最新修改；

$ git merge --no-ff -m "merge with no-ff" dev

合并分支时，加上--no-ff参数就可以用普通模式合并，合并后的历史有分支，能看出来曾经做过合并，而fast forward合并就看不出来曾经做过合并。

查看远程库信息，使用git remote -v；

本地新建的分支如果不推送到远程，对其他人就是不可见的；

从本地推送分支，使用git push origin branch-name，如果推送失败，先用git pull抓取远程的新提交；

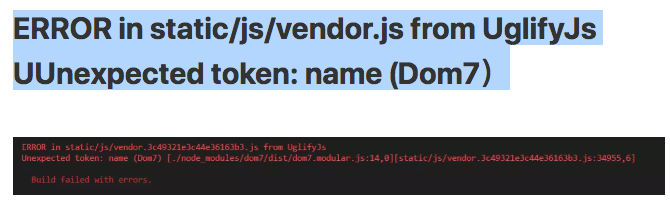
在本地创建和远程分支对应的分支，使用git checkout -b branch-name origin/branch-name，本地和远程分支的名称最好一致；

建立本地分支和远程分支的关联，使用git branch --set-upstream branch-name origin/branch-name；

从远程抓取分支，使用git pull，如果有冲突，要先处理冲突。

git push -f:强制覆盖远程代码

2，解决打包的时候报错uglify不能编译es6解决方案，（eg:element-ui中使用了es6 的语法，UglifyJs无法对 es6进行混淆压缩。



(1).在项目中找到 build/webpack.base. config.js文件

(2)、打开文件，把 module 的 rules 的 test: /.js$/做出修改

// 修改前

rules: [

test: /\.js$/,

loader: 'babel-loader',

include: [resolve('src'), resolve('test')]

]

// 修改后

rules: [

...

include: [resolve('src'), resolve('test'),resolve('/node\_modules/element-ui/src'),resolve('/node\_modules/element-ui/packages')]

]

// 即把element-ui框架的源码进行 es6到 es5的转化

3.使用jquery,需要先安装，再配置,使用时引入

npm i jquery -S;

webpack.base. config.js中配置：

plugins: [

new webpack.ProvidePlugin({

$:'jquery',

jquery:"jquery",

jquery:"jquery",

"window.jquery":"jquery"

})

]

5.封装请求：

export function fetchCardUpdate({ commit }, {repositoryId, cardNo}) {

return (async () => {

try {

const params = {

repositoryId,

cardNo

};

// 改变loading状态为true

commit('setLoading', true, {root: true});

const response = await load({

method:'post',//不写默认为get;

url: '/devsecop/index.php/card/update',

data: params

});

// 改变loading状态为false

commit('setLoading', false, {root: true});

return Promise.resolve(response);

} catch (err) {

// 改变loading状态为false

commit('setLoading', false, {root: true});

return Promise.reject(err);

}

})();

}

// 删除项目(csrf)

export function deleteProject({commit}, {id, \_csrf}) {

return (async () => {

try {

const params = {

id,

\_csrf

};

if (!\_csrf) {

const csrf = await load({

url: '/project/web/index.php/site/gettoken'

});

params.\_csrf = csrf.\_csrf;

}

const response = await load({

url: '/project/web/index.php/item/delitem',

data: params,

method: 'post',

json: false

});

return Promise.resolve(response);

} catch (err) {

return Promise.reject(err);

}

})();

}

1. 解决浏览器回退得缓存问题，刷新页面

created(){

window.addEventListener('popstate', function(event) {

window.location.reload();

})

或

window.addEventListener('hashchange', function(event) {

window.location.reload();

})

},

1. vue使用轮播图：
2. 安装
3. import 'swiper/dist/css/swiper.css'
4. import Swiper from 'swiper';
5. mounted(){

new Swiper ('.swiper-container', {

loop: true,

autoplay: {

delay: 3000,

stopOnLastSlide: false,

disableOnInteraction: true,

}

})

this.load();//加载首页的数据

$('.backtop').hide();

window.addEventListener('scroll', this.handleScroll)

},

1. 回到顶部

handeltop(){

$('.backtop').on('click',function(){

$('html,body').animate({scrollTop:$('html,body').offset().top}, 1000);

//$("html,body").animate({scrollTop:0},400);

})

}

border-collapse: collapse;

border-spacing: 0;

1. 动画rain

@-webkit-keyframes rain

{

0%

{

opacity: 0;

}to

{

-webkit-transform: translate3d(0,0,0);

transform: translate3d(0,0,0);

opacity: 1;

}

}@-moz-keyframes rain

{

0%

{

opacity: 0.5;

}to

{

-moz-transform: translate3d(0,0,0);

transform: translate3d(0,0,0);

opacity: 1;

}

}@-o-keyframes rain

{

0%

{

opacity: 0.5;

}to

{

transform: translate3d(0,0,0);

opacity: 1;

}

}@keyframes rain

{

0%

{

opacity: 0.5;

}to

{

-webkit-transform: translate3d(0,0,0);

-moz-transform: translate3d(0,0,0);

transform: translate3d(0,0,0);

opacity: 1;

}

}

动画进度条

 animation: progressIn 5s linear both;

@keyframes progressIn{

    from{width:0;}

    to{width:100%;}

}

1. Cccc

// 隐藏退出操作

            hideUserOpt: function(ele) {

                var $useropt = $('#useropt');

                if (ele.parents('#useropt').length < 1 && $('#userinfo').hasClass('hide')) {

                    // $useropt.addClass('hide').siblings().removeClass('hide');

                    $useropt.animate({

                        opacity: 0,

                        height: '42px'

                    }, 250, function(){

                        $useropt.addClass('hide');

                        $useropt.siblings().removeClass('hide')

                                .animate({

                                    opacity: 1

                                }, 250);

                    })

                }

            },

1. keyenter,keydown

Keyenter:回车 keydown:按下键盘就触发

<input type="text" v-model="hbl" placeholder="请输入查找人邮件前缀，例如：zhangsan" class="item-input" @keyup.enter="handleSearch" @keydown="first=true;">

1. 监听到值hbl变化就打印111；

https://www.imooc.com/article/70010

watch:{

"hbl"(val){

console.log(111);

},

}

14.导入导出

//确定下载导出pdb信息

handleExportPDB() {

// 默认要导出的pdb是新版的

let url = '/mytool/web/index.php/pdb/newpdbexcel'

let \_self = this;

\_self.$ajax.get('/site/gettoken')

.then(function(res){

let data = {

\_csrf: res.data.token,

pdbl: \_self.pdbType

}

// 模拟表单

let form = document.createElement('form')

form.action = url

form.method = 'post'

// 添加表单参数

for(var key in data) {

if (data.hasOwnProperty(key)) {

let input = document.createElement('input')

input.name = key

input.value = data[key]

input.type = 'hidden'

form.appendChild(input)

}

}

// 添加表单到body

document.body.appendChild(form)

// 提交表单

form.submit()

// 删除表单

if (form.remove) {

form.remove()

} else {

form.parentNode.removeChild(this)

}

// 关闭弹窗

\_self.load = !\_self.load;

})

.catch(function(error){

this.load= !this.load;

})

},

1. 省略号

Overflow: hidden;

White-space: nowrap;

Text-overflow: ellipsis;

多行文本溢出显示省略号实现方法：

display: -webkit-box;

-webkit-box-orient: vertical;

-webkit-line-clamp: 2;

text-overflow: ellipsis;

overflow:hidden;

1. :title=’谢谢谢谢谢’

17.[vue-axios基本用法](https://www.cnblogs.com/silent007/p/8603367.html)

首先安装axios：

1)：npm install

2)：npm install vue-axios --save

3)：npm install qs.js --save　　//它的作用是能把json格式的直接转成data所需的格式

安装成功后，在main.js页面引用：

import Vue from 'vue'

import axios from 'axios'

import qs from 'qs'

Vue.prototype.$axios = axios //全局注册，使用方法为:this.$axios

Vue.prototype.qs = qs //全局注册，使用方法为:this.qs

最后开始使用请求：

<script>

export default{

data(){

return{

userId:666,  
　　　　　　　　　 token:'',

}

},

created(){

this.$axios({

method:'post',

url:'api',

data:this.qs.stringify({ //这里是发送给后台的数据

userId:this.userId,

token:this.token,

})

}).then((response) =>{ //这里使用了ES6的语法

console.log(response) //请求成功返回的数据

}).catch((error) =>{

console.log(error) //请求失败返回的数据 })

}

}</script>

1. Echarts
2. 官网
3. <https://echarts.baidu.com/echarts2/doc/example.html>--(hl)

19.

function page() {

if ((navigator.userAgent.match(/(phone|pad|pod|iPhone|iPod|ios|iPad|Android|Mobile|BlackBerry|IEMobile|MQQBrowser|JUC|Fennec|wOSBrowser|BrowserNG|WebOS|Symbian|Windows Phone)/i))) {

//移动端

}

else {

//PC端

}

}

page();

20.// 获取当前滚动条高度

getScrollTop(){

var scrollTop = 0 ; if(document.documentElement&&document.documentElement.scrollTop){

scrollTop = document.documentElement.scrollTop;

} else if(document.body){

scrollTop = document.body.scrollTop;

}

return scrollTop;

},

21.

/\*hasOwnProperty() 方法会返回一个布尔值，指示对象自身属性中是否具有指定的属性\*/

/\*语法obj.hasOwnProperty(prop)\*/

var data = {

hasOwnProperty : function() {

return false;

} ,

name : "Jackey"

}

1. vue数组变异解决: vue.set(target,key,val);

Tip:Vue.set()在methods中也可以写成this.$set()

调用方法：Vue.set( target, key, value )

target：要更改的数据源(可以是对象或者数组)

key：要更改的具体数据

value ：重新赋的值

23.格式化时间操作

formateTime(date) {

return date.getFullYear() + '-' + ((date.getMonth() + 1) > 9 ? date.getMonth() + 1 : '0' + (date.getMonth() + 1)) + '-' + (date.getDate() > 9 ? date.getDate() : '0' + date.getDate());

}

24.关于webpack配置问题

<https://www.cnblogs.com/whkl-m/p/6627864.html>

1. 防抖和节流

（1）、节流（规定时间内只能操作一次）JS:

let oBtn=document.getElementById('show');

let oBox=document.getElementById('box');

/\*

handle:buy函数

wait:规定在一秒钟内只能执行一次

\*/

function throttle (handle, wait) {

let lastTime = 0;

return function (e) {

let nowTime = new Date().getTime()

if (nowTime - lastTime > wait) {

handle();

lastTime = nowTime;

}

}

}

function buy(){

oBox.innerText = parseInt(oBox.innerText)+1

}

oBtn.onclick = throttle(buy, 1000)

（2）、解决节流（比如百度搜索框不会每次输入一个字就发请求）

let oIpt = document.getElementById('ipt');

let time = null;

function debounce (handle, delay) {

let time = null;

return function () {

let self = this,arg = arguments;

clearTimeout(time);

time = setTimeout(function () {

handle.apply(self,arg);　　//this绑定

},delay)

}

}

function ajax (e) {

console.log(e,this.value)

}

oIpt.oninput = debounce(ajax, 1000) //1s后发出请求

1. 获取近七天的方法（formatetime是格式化时间为2019-09-12格式）

const end = new Date()

const start = new Date()

start.setTime(start.getTime() - 3600 \* 1000 \* 24 \* 7)

var ti = [this.formateTime(start), this.formateTime(end)];

console.log(ti);

1. 搭建本地php环境

<https://blog.csdn.net/qq_36595013/article/details/80373597>

<https://blog.csdn.net/qq_36595013/article/details/80373597>

nginx

28..editorconfig 的文件。该文件的内容定义该项目的编码规范。

EditorConfig 支持的常用的编码规范，如下

https://github.com/editorconfig/editorconfig/wiki/EditorConfig-Properties

1.charset：文件编码。可选值

latin1

utf-8。一般用这个。

utf-16be

utf-16le

2.indent\_style: 缩进类型。可选值

space

tab

3.indent\_size: 缩进数量。可选值

整数。一般设置 2 或 4。

tab

4.insert\_final\_newline：是否在文件的最后插入一个空行。可选值

true

false

5.end\_of\_line：换行符格式。说明见Wiki：换行。可选值

lf。一般用这个。

crlf

cr

6.trim\_trailing\_whitespace：是否删除行尾的空格。可选值

true

false

# 29.(1).vue项目结构及文件介绍(2) —— build文件夹

<https://blog.csdn.net/luwenze/article/details/89430527>

(2).Config/index.js配置

https://www.cnblogs.com/wulinzi/p/8066726.html

1. localStorage的使用

localStorage.getItem(key):获取指定key本地存储的值  
 localStorage.setItem(key,value)：将value存储到key字段

localStorage.setItem(key,JSON.stringify(value))：将value存储到key字段



解决本地存储的值改变的时候js不能实时监听到的问题

问题描述：我们在js里面获取了某一个localstorage的值，但是后期它可能改变了，我们js只执行一遍没办法再次获取它的值，当然可以刷新页面获取，但如果是我们的但页面就不能刷新页面了，此时：

我们可以重写localStorage的setItem方法，当调用setItem方法设置新值的时候，会new Event('setItemEvent')

但是只是实例化事件了，怎么去监听呢？

我们可以用window.dispatchEvent()这个方法来派发一个事件，让window去监听。新实例出来的这个事件会被绑定到window上来监听它的调用，当调用的时候可以拿到setItem的新值，直接赋值就可以

【 操作url获取参数的：】

let url = '?wd=蔡徐坤&skill=篮球&year=2019'; let searchParams = new URLSearchParams(url); for (let p of searchParams) { console.log(p); }  
 链接：<https://juejin.im/post/5d038c9051882548ac439933>

get、has、append、delete

Eg:searchParams.append('age', 26);

searchParams.set('skill', '篮球 唱 跳 rap');

searchParams.toString();



点击的时候，切换slider,重新设置到localstrage里面去

updated() {

this.changeSidebar()

},

31.注：qs，使用axios，必须得安装 qs，所有的Post 请求，我们都需要 qs,对参数进行序列化。

# 32.**vue-axios interceptors(拦截器)实际应用**

<https://www.jianshu.com/p/ff8541e0976a>

([axios 拦截 , 页面跳转, token 验证](https://www.cnblogs.com/dhsz/p/6410031.html))

https://www.cnblogs.com/dhsz/p/6410031.html

33.

if(/phone|pad|pod|iPhone|iPod|ios|iPad|Android|Mobile|BlackBerry|IEMobile|MQQBrowser|JUC|Fennec|wOSBrowser|BrowserNG|WebOS|Symbian|Windows Phone|micromessenger/i.test(navigator.userAgent)) {

this.isMobile=true;

1. 监听对象和属性

watch: {

addForm: {

handler: function() {

// 每次更新时，重新校验规则

if (this.firsted) {

this.handleValidateForm()

}

},

deep: true

},

'addForm.question\_type': function(val){

if(this.businessManager){

return;

}

if(val=='501'){

this.getManager()

}

}

}

1. 慕课网：

1).axios视频: http://www.imooc.com/learn/1152

2).web安全xss攻击防范： http://www.imooc.com/learn/812

1. 表格td设置宽度不生效
2. table{table-layout: fixed;word-break: break-all; word-wrap: break-word; //表格固定布局
3. td{

width:100px;

white-space: nowrap;

text-overflow: ellipsis;

overflow:hidden;

cursor: pointer;

//超出部分显示省略号,并给td一个:title=’ssssss’

}

1. 高度塌陷：

（1）什么是高度塌陷？==》父元素高度自适应，子元素 float 后，造成父元素高度为0，称为高度塌陷问题。

（2）如何来解决高度塌陷？

方案一：给父元素一个固定的高度

缺点：给父元素固定高度违背了高度自适应的原则，不够灵活，不推荐使用。

方案二：给父元素添加属性 overflow: hidden;

优点：浏览器支持好，简单；

缺点：当子元素有定位属性时，设置 overflow: hidden; 容器以外的部分会被裁剪掉

方案三：在子元素的末尾添加一个空的 div ，并设置下方样式

div{

clear: both;

height: 0;

overflow: hidden;

}

优点：所有浏览器都支持，并且容器溢出不会被裁剪；

缺点：在页面中添加无意义的div，容易造成代码冗余。

方案四：万能清除浮动法

在父元素中内容的最后添加一个伪元素来实现第三种方案的功能，具体设置样式如下：

父元素:after{

content: "";

height: 0;

clear: both;

overflow: hidden;

display: block;

visibility: hidden;

}

优点：不会造成代码冗余，剩余代码性能优化，推荐使用。

1. 常见正则验证:

<https://blog.csdn.net/xuemoyao/article/details/8033138>

https://juejin.im/post/5cdcd42551882568651554e6（正则学习）

(1)汉字：/^[\u4e00-\u9fa5]{0,}$/

(2)日期格式：^\d{4}-\d{1,2}-\d{1,2}

(3)\bOlive\b 匹配单词Olive 例如： I Love Oliver and Olive .这个时候返回的是Olive 而不是Oliver,因为\b....\b返回的匹配的单词,匹配边缘

(4)需要匹配qq邮箱 \d{8,}+qq+\.+com  在这里的. 就需要加斜杠

1. a标签想要点击

href:”javascript:;”//防止产生空链接的作用

div[width].box1#box2{div标签内的innerhtml}

div>img

添加父容器，选中，control+shift+p，输入wrap后选择wrap with apprev..

回车写div.box

text-indent：20px;//将文字向右挤下20px;不当住背景图

1. 什么时候需要用的Vue.nextTick()？？

1、Vue生命周期的created()钩子函数进行的DOM操作一定要放在Vue.nextTick()的回调函数中，原因是在created()钩子函数执行的时候DOM 其实并未进行任何渲染，而此时进行DOM操作无异于徒劳，所以此处一定要将DOM操作的js代码放进Vue.nextTick()的回调函数中。与之对应的就是mounted钩子函数，因为该钩子函数执行时所有的DOM挂载已完成。

created(){

let that=this;

that.$nextTick(function(){ //不使用this.$nextTick()方法会报错

that.$refs.aa.innerHTML="created中更改了按钮内容"; //写入到DOM元素

});

},

1. 当项目中你想在改变DOM元素的数据后基于新的dom做点什么，对新DOM一系列的js操作都需要放进Vue.nextTick()的回调函数中；通俗的理解是：更改数据后当你想立即使用js操作新的视图的时候需要使用它

正确的用法是：vue改变dom元素结构后使用vue.$nextTick()方法来实现dom数据更新后延迟执行后续代码

3、在使用某个第三方插件时 ，希望在vue生成的某些dom动态发生变化时重新应用该插件，也会用到该方法，这时候就需要在 $nextTick 的回调函数中执行重新应用插件的方法。

1. 给HTML设置lang='zh-CN'

是一个语言功能。

如果是中文网站设置zh-CN；如果是英文网站设置en;

主要是告诉搜索引擎，此网站使用的语言版本。

英文: <html xmlns=xml:lang="en" lang="en" >

简体中文: <html xmlns= xml:lang="zh-CN" lang="zh-CN">

1. 使用markdown

<https://github.com/hinesboy/mavonEditor>

https://www.fontke.com/tool/fontface/

https://www.fontke.com/tool/subfont/

字体生成工具，字体提取工具。

1. 真机调试

Mac:ifconfig 找到IP地址，IP：端口号访问

1. 网易课程之（如何开发组件库级别组件）分析源码
2. 业务组件与通用组件

提取业务功能与基本功能，实现基本功能，定义业务功能接口

slot使用，自定义的内容

Create: 数据准备好

Mounted：dom准备好

Eg：位置宽度的盒子，想让他居中另外方法：

Top:50%;left:50%;

然后mounted里面操作，获取dom生命周期

Eg:

<div ref="con" :style="constyle">dddd</div>

mounted(){

this.constyle={

marginLeft:-this.$refs.con.clientWidth/2+"px",

marginTop:-this.$refs.con.clientHeight/2+"px",

}

}

1. 使用小妙招：不使用$emit，也不使用发布订阅模式，也不用bind

<pop :fater=”this”></pop>

pop组件用prop接收father,这样就可以拿到this，相当于父组件对象

1. 比如父传子，传一个父组件loading()方法过去,子组件调用

<pop :fater=”this” :load=”loading” v-if=”pop”></pop>

Prop:[“load”]

This.load() x

This.load.call(“this.father”) √

或者

<pop :fater=”this” :load=”loading.bind(this)”></pop>

This.load()

1. 代码健壮性考虑：

比如：弹窗组件有的地方有调用父组件canclehandel方法，有的就仅仅关闭页面

Cancle(){

//给予默认值，防止没有调用该方法报错

Var \_fn=this.canclehandel||function(){};

\_fn.call(this.father);

This.fahter.pop=false

}

1. 代码简洁性处理

接口隔离模式：细化接口，方便可扩展性

外观模式提供更高的统一接口

1. instanceof和typeof
2. 网易课程之 （深度剖析spa前端路由原理与实现方式）

实现前端路由的两种机制hash和history

（1）hash

hash的变化：不会导致浏览器向服务器发起请求；

hash的改变会触发hashchange实件；

H5的history出现之前，基本都是使用hash模式来实现我们的前端路由

//我们可以通过hashchange事件捕捉到URL的变化，通过这个特性去事项我们的Spa 路由

Js:

/\*

1. 创建一个路由对象，register方法，用于处理每个hash值对应的回调
2. 不存在hash的时候，默认是首页，实现一个index方法用于处理首页的回调
3. 通过hashchange事件去监听我们hash的变化，定义hash变化的回调

\*/

Function Router(){

//用于存储不同的hash值对应的回调函数

This.routes={};

//监听hasjchange事件

Window.addEventListener(“hashchange”,this.reload());

}

//首页的回调

Router.prototype.index=function(callback){

This.routers[‘index’]=callback;

}

//其他页面的回调

Router.prototype.register=function(hash,callback){

This.routers[hash]=callback;

}

//实现reload方法

Router.prototype.reload=function(){

Let callback；

Let hash=location.hash.slice(1)

!hash ? callback=This.routers.index :

callback= This.routers[hash];

}

1. History

H5中两个新的API,改变url，不会去刷新页面，具备前端路由的能力

History.pushState();

History.replaceState();

【总结】：

a、hash改变，可以通过hashchange进行监听

b、history的改变，不会触发任何时间，所以我们无法监听history，我们可以罗列所有的可能触发history改变的情况，把这些情况进行一个拦截，变相的去监听history的变化

1. 点击浏览器的前进后退
2. 点击a标签
3. js中，触发history.pushState()函数
4. js中，触发History.replaceState()函数

（1）全局配置公用

//eg: 将axios挂载到prototype上，在组件中可以直接使用this.axios访问

Vue.prototype.axios = axios;

Vue.prototype.conf=conf;

Vue.prototype.csrf='';

1. key值可为[types.STATES]
2. 不使用var，直接给定义变量，隐式的声明了全局变量

tt='test';

function ff(){

alert(tt);

}

这种方法需要注意的地方：如果是在一个函数体内去定义变量的话，只有当该函数执行后它才变成了全局变量，才能被其他函数知道，所以最好定义在函数体外

router.beforeEach((to, from, next) => {

// 统计代码

if (to.path) {

\_hmt.push(['\_trackPageview', '/#' + to.fullPath]);

}

next();

});

注意在方法外不加var定义变量需对其进行赋值，否则出现xx is not defined）

a.都加var，在方法内则是局部变量，在方法外则是全局变量。

b.在方法内,加var为局部变量，不加var则是全局变量（在执行当前方法之后）

(3)favicon.ico，是网站的一个小标志

A.首先将favicon.ico图像准备好，通常有两个尺寸，32PX或者64PX的正方形

B.新建一个空白的HTML文档，用于装载FAVICON图标

C. 在<head>标签之间，插入favicon.ico引入语句，示例：

<link rel="shortcut icon" href="/favicon.ico" type="image/x-icon" />

D.将favicon.ico文件上传到网站的根目录，

E.对于现代的浏览器，如果将favicon.ico文件放在了网站的根目录，是可以选择不写ico文件引入文件的，浏览器是自动请求该文件的

（4）vue项目中通过 @ 引入文件，可以在webpack.base.conf.js文件中

resolve: {

extensions: ['.js', '.vue', '.json'],

alias: {

'vue$': 'vue/dist/vue.esm.js',

'@': resolve('src'),

'@utils': resolve('src/utils'),

'@projManage': resolve('src/components/projectManage'),

}

}

　　配好后就可以在项目中使用了

如：需要引入 src/components/projectManage/projectList.vue文件，就可以写成：import projectList from '@projManage/projectAdmin/projectList'

1. 网易课程之（前端高性能优化最佳实践）

页面加载性能、动画与操作性能、内存电能消耗

1. vue打包优化
2. 网易课程之 （源码阅读jquery）
3. canvas学习：
4. context.clearRect(x,y,width,height);

Eg1:// 先绘制图片

var img = new Image();

img.onload = function () {

context.drawImage(img, 0, 0, 250, 167);

// 中间开个方形的洞

context.clearRect(50, 50, 100, 66);

};

img.src = './1.jpg';

// 登录状态下不会出现这行文字，点击页面右上角一键登录

Eg2:慢动作演示动画刷新绘制

context.font = '16px STHeiti, SimHei';

// 圆球的水平位置和半径

var x = 30, radius = 30;

// 清除方法

var clear = function () {

// 清除画布

context.clearRect(0, 0, 250, 167);

context.fillText('清除', 20, 30);

setTimeout(draw, 500);

};

// 绘制球方法

var draw = function () {

x += 10;

if (x > 220) {

x = 30;

}

// 清除提示文字

context.clearRect(0, 0, 250, 167);

// 新的文字提示，以及圆绘制

context.fillText('绘制', 20, 30);

context.beginPath();

context.arc(x, 84, radius, 0, Math.PI \* 2);

context.fillStyle = '#039';

context.fill();

// 清除

setTimeout(clear, 500);

};

// 初始触发

draw();

1. SwitchHosts，host切换神器

<https://oldj.github.io/SwitchHosts/>

1. [js去除字符串空格(空白符)](https://www.cnblogs.com/a-cat/p/8872498.html)

使用js去除字符串内所带有空格，有以下三种方法：

( 1 ) replace正则匹配方法

去除字符串内所有的空格：str = str.replace(/\s\*/g,"");

去除字符串内两头的空格：str = str.replace(/^\s\*|\s\*$/g,"");

去除字符串内左侧的空格：str = str.replace(/^\s\*/,"");

去除字符串内右侧的空格：str = str.replace(/(\s\*$)/g,"");

( 2 ) str.trim()方法

　　trim()方法是用来删除字符串两端的空白字符并返回，trim方法并不影响原来的字符串本身，它返回的是一个新的字符串。

缺陷：只能去除字符串两端的空格，不能去除中间的空格

var str = " 6 6 ";

var str\_1 = str.trim();

( 3 ) JQ方法：$.trim(str)方法

　　$.trim() 函数用于去除字符串两端的空白字符。

　　注意：$.trim()函数会移除字符串开始和末尾处的所有换行符，空格(包括连续的空格)和制表符。如果这些空白字符在字符串中间时，它们将被保留，不会被移除。

var str\_1 = $.trim(str);

# 

1. Mac:

开启mac自带的apache和php的支持https://www.jianshu.com/p/613e775dab22

apache显示目录，建一个.htaccess写入

<Files \*> Options Indexes </Files>

1. vim 学习：https://www.cnblogs.com/dalaoban/p/9381305.html
2. 代码对比工具下载与使用：<https://www.scootersoftware.com/download.php>
3. favicon

[强制] 保证 favicon 可访问。

解释：在未指定 favicon 时，大多数浏览器会请求 Web Server 根目录下的 favicon.ico 。为了保证 favicon 可访问，避免 404，必须遵循以下两种方法之一：

在 Web Server 根目录放置 favicon.ico 文件。

使用 link 指定 favicon。

示例：

<link rel="shortcut icon" href="path/to/favicon.ico">

编码规范

（1）<http://gitlab.baidu.com/fe/spec/blob/master/vue.md>

（2<https://github.com/ecomfe/spec/blob/master/html-style-guide.md>

（3<https://github.com/ecomfe/spec/blob/master/css-style-guide.md>

（4<https://github.com/ecomfe/spec/blob/master/javascript-style-guide.md>

58、php网易通关视频：

https://study.163.com/course/courseLearn.htm?courseId=1005641004#/learn/video?lessonId=1052893262&courseId=1005641004

一个合格(优秀)的前端都应该阅读这些文章

https://juejin.im/post/5d387f696fb9a07eeb13ea60

1. 安装nvm

https://www.cnblogs.com/feckl/p/8013233.html

# 解决nvm安装成功后，但命令不可用(command not found)

1、进入nvm安装到文件目录

     cd ～/.nvm

2、查看目录下文件列表

     ls -a

3、若无.bash\_profile文件，则创建该文件：

创建文件步骤：

3.1 touch .bash\_profile

3.2 open .bash\_profile

3.3 将下部分文件粘贴至 .bash\_profile文件

export NVM\_DIR="/Users/jack-le/.nvm"

[ -s "$NVM\_DIR/nvm.sh" ] && . "$NVM\_DIR/nvm.sh" # This loads nvm

（此语句是配置文件，与自己电脑有关，若不知道，可运行如下命令，会提示这段语句）

curl -o- https://raw.githubusercontent.com/creationix/nvm/v0.25.4/install.sh | bash

3.4 保存文件，关闭此文件

3.5 更新刚配置的环境变亮

      source .bash\_profile

1. 输入nvm命令，验证是否成功，
2. Nvm ls查看已经使用过的node版本

例如nvm use 6.10.2，若已安装node 该版本node，则提示已切换至此版本，

若无安装，则提示未安装此版本，请安装。

此时，nvm可正常使用

60，关于滚动

隐藏滚动条

::-webkit-scrollbar{display: none;}

滚动条属性：(不会传递给父元素)

Overflow-behavior: cantain

1. safari获取时间戳兼容问题：

Eg: var time = “2020-02-11 12:00:00”

New Date(time.replace(/-/g, ‘/’)).getTime();