本手册仅记录使用vue-cli 2.0搭建vue项目的使用技巧和规则。

# 安装环境

1. node.js安装：<https://nodejs.org/en/download/> 软件下载安装即可
2. webpack安装：npm install webpack -g
3. vue-cli 2.x安装：npm install -g vue-cli

# 创建项目

命令：vue init webpack 项目名

# Vue命令

## v-bind 单向绑定

动态更新HTML元素上的属性。

<a **v-bind:href**="URL">连接地</a>

<button @click="URL = url[0]">百度</button>

<button @click="URL = url[1]">京东</button>

<button @click="URL = url[2]">淘宝</button>

data() {

return {

URL:``,

url:[`https://www.baidu.com`,`https://www.jd.com`,`https://www.taobao.com`]

}

}

### CSS绑定

**Class模式**

<el-input v-model.lazy="dat2" size="normal" clearable :class="{wid:flag}"></el-input>

data() {

return { flag:true };

}

CSS

<style scoped>

.wid { width: 200px; }

</style>

**Style模式**

<p :style="{fontSize:**num**+'px'}">变化的字体</p><!--style模式下不可以用xxx-xxx形式，而是用驼峰命名-->

## v-model双向绑定

常用于输入框。~.lazy 失焦后更新，~.number 转换为数字，~.trim 前后清空。

<input type="text" v-model.trim="URL\_NAME"/>

data() {

return {

URL\_NAME:``,

}

}

## v-for循环创建DOM

v-for即可遍历数字，也可遍历对象数组，使用v-for时:key必须绑定一个唯一值。

<div v-for="(item,index) in 10" :key="index">{{item}}</div>

## v-if控制DOM显现/消失

<div v-if="show">展示</div>

<div v-else>不展示</div>

<button @click="show=!show">展示</button>

## v-else / v-else-if

配合v-if进行if...else...判断。

## v-on

### 阻止事件冒泡

v-on:click.stop="..."阻止事件冒泡

v-on:prevent="..."阻止原生事件

# Vue方法

## datas数据域

定义格式为函数形式，如下：

**methods方法**

data() {

return {

...

};

}

Methods中定义一切用于复杂计算和调用后台的方法。Methods中的方法最好用funName(){}格式，箭头函数可能会导致意想不到的错误。

methods: {

...

}

## computed计算属性

相当于动态返回数据类型的方法；有两种定义方法；用法和data一样。

computed:{

//第一种定义方式，也可传参，参数指向this

dat1\_dat2(){

return `${this.dat1}-${this.dat2}`

},

//第二种定义方式，具备get/set方法

dat1\_dat2:{

get(){

return this.swap

},

set( newVal ){ this.swap = newVal }

}

}

## watch监视器

watch可监视data中的数据。默认function(newValue,oldValue){}和function(newValue){}

watch:{

dataValue: function(newValue,oldValue){

...

}

...

}

## filter过滤器

function格式定义在filter的标签中，使用在{{}}值的尾部添加管道符“|”对数据进行过滤，常用于格式化文本。过滤器也可以串联，

**动态过度**

{{name|toUpper}}

data() {

return {

name:`jAcK`

}

},

filters:{

      toUpper(value){

          return value.toUpperCase()

      }

}

## 生命周期

new Vue()

Init Events & Lifecycle

Init injections & reactivity

**beforeCreate**

Has el option?

Has template option?

**YES**

**created**

**NO**

**When vm.$mount(el) is called**

Create vm.$el and replace el with it

Compile el’s outerHTML as template

Compile template into render function

**beforeMounted**

Mounted

**mounted**

Teardown watchers, child components and event listeners

**When vm.$destroy() is called**

**Destroyed**

**destroyed**

**beforeDestroyed**

Virtual DOM re-render and patch

**when data changes**

**beforeUpdate**

**Updated**

**YES**

**NO**

Mounted():发送ajax请求，如定时器等异步任务

beforeDestroy:清除定时器

根据自己需要创建项目。

组件

引用

# 组件间数据传递

父传子

子传父

中间层