# Vue基础

## vue基础知识

### 内置指令

#### 指定标签体

- v-text ： 当作纯文本
- v-html ： 将value作为html标签来解析

#### 显示/隐藏元素

- v-if ： 如果vlaue为true， 当前标签会输出在页面中
- v-else ： 与v-if一起使用， 如果value为false， 将当前标签输出到页面中
- v-show: 就会在标签中添加display样式， 如果vlaue为true， display=block， 否则是none

#### 遍历

- 遍历数组 ： v-for="(person, index) in persons"
- 遍历对象 ： v-for="value in person"  $key

#### 绑定事件监听

- v-on:事件名， 可以缩写为: @事件名
- 监视具体的按键: @keyup.keyCode  @keyup.enter
- 停止事件的冒泡和阻止事件默认行为: @click.stop  @click.prevent
- 隐含对象: $event

#### 强制绑定解析表达式

- v-bind:属性， 可以缩写为: :属性名

#### v-model

- 双向数据绑定
- 自动收集用户输入数据

#### 标识某个标签

- ref='xxx'
- 读取得到标签对象: this.$refs.xxx

### 组件

- 定义组件:

```
<script>
    Vue.component("con1",{
        props:{

        },
        template:""
});
</script>
```

- 使用组件:

```
<div id="vue">
    <con1></con1>
</div>
```

## axios

> 作用: 用于网络通信, 相当于ajax
> 官网: http://axios-js.com/zh-cn/docs/vue-axios.html

引入

```
方式一:
    npm install --save axios vue-axios
    然后在maxin.js引用Axios
    import axios from 'axios'
    import VueAxios from 'vue-axios'
    Vue.use(VueAxios, axios)
方式二:
    <script src="https://unpkg.com/axios/dist/axios.min.js"></script>
```

用法:

```
// 1: 引入
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
// 1-2:或者执行
// 2:调用方法
<script>
    ...
    data() {
        return {
            info:{}
        }
    },
    methods:{
        getData:function() {
            axios.get("./data.json").then(response => {
                this.info = response.info;
            });
```

```
        }
    }
    ...
</script>
```

## 计算属性

- 什么是计算属性：

  本质上是一个属性，通过执行方法把得出的数据赋给属性。与普通方法不同的是：它在**内存**中，计算一次后不会轻易改变，
除非里面的属性发生了变化，才会重新执行一次方法。

- 有什么优点：

  可以减少浏览器内存的消耗

- 用于什么场合：

  **一般用于不会经常变动的属性上。**

- 写法

```
...
computed() {
    calCount: function() {
        return 100+20;
    }
}
...
```

## 插槽

## 自定义事件内容分发

理解：

其实说白了，就是子组件如何调用父组件的方法，或者说是 **组件通信**

关键点：this.$emit('方法名',param);别忘了:key

例子：

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>slot</title>
    <script src="https://cdn.jsdelivr.net/npm/vue@2/dist/vue.js"></script>
</head>
<body>
    <div id="vue">
```

```
        <tech>
            <tech-title slot="tech-title" :title="title"></tech-title>
            <tech-list slot="tech-list" v-for="(item,idx) in items"
                       :item="item" :index="idx" :key="idx"
@remove="removeItems(idx)" ></tech-list>
        </tech>
    </div>

    <script>
        Vue.component("tech",{
            template:'<div>\n' +
                '            <slot name="tech-title"></slot>\n' +
                '            <ul>\n' +
                '                <slot name="tech-list"></slot>\n' +
                '            </ul>\n' +
                '        </div>'
        });
        Vue.component("tech-title",{
            props:["title"],
            template: '<div>{{title}}</div>'

        });
        Vue.component("tech-list",{
            props: ["item","index"],
            template:'<li>{{index}}-----{{item}} <button @click="remove">
删除</button></li>',
            methods:{
                remove:function (index) {
                    this.$emit('remove',index);
                }
            }
        });

        var vue = new Vue({
            el:"#vue",
            data() {
                return{
                    title:"技术栈",
                    items:["mysql","redis","zookeeper","vue","elementUI"]
                }
            },
            methods: {
                removeItems:function (index) {
                    // alert("要删除的元素: "+this.items[index]);
                    this.items.splice(index,1);
                }
            }
        });
    </script>
```

```
</body>
</html>
```

步骤：

点击删除按钮如何删除对应item

1、tech-list组件如何调用自己组件中的删除方法

2、vue里如何调removeItems方法

3、tech-list组件中删除方法(remove)如何删除vue里的removeItems方法

# 搭建vue脚手架工程

## 安装node.js

1). 下载并安装

官网：https://nodejs.org/zh-cn/

安装：一路next即可

查看是否成功：

```
C:\Users\97146>node -v
v14.15.4
```

2). npm包管理器，是集成在node中的，所以，直接输入**npm -v**就会显示出npm的版本信息

```
C:\Users\97146>npm -v
6.14.10
```

```
### 安装cnpm
```

> 由于npm的有些资源被墙，为了更快更稳定,所以我们需要切换到淘宝的npm镜像——cnpm。

1). cnpm网站

https://npmmirror.com/

2). 安装

```
npm install -g cnpm --registry=https://registry.npm.taobao.org
```

3). 查看是否安装完成

cnpm -v

```
C:\Users\97146>cnpm -v
cnpm@6.1.1 (C:\Users\97146\AppData\Roaming\npm\node_modules\cnpm\lib\parse_argv.js)
npm@6.14.11 (C:\Users\97146\AppData\Roaming\npm\node_modules\cnpm\node_modules\_npm@6.14.11@npm\lib\npm.js)
node@14.15.4 (D:\software\install software\work\nodejs\node.exe)
npminstall@3.28.0 (C:\Users\97146\AppData\Roaming\npm\node_modules\cnpm\node_modules\_npminstall@3.28.0@npminstall\lib\index.js)
prefix=C:\Users\97146\AppData\Roaming\npm
win32 x64 10.0.19042
registry=https://r.npm.taobao.org
```

4). 使用cnpm

> 使用cnpm的方法就是，需要用到npm的地方直接使用cnpm替换就可以了

## 安装vue

```
cnpm install vue
```

## 安装vue-cli脚手架构建工具

简介:

vue-cli 提供一个官方命令行工具，可用于快速搭建大型单页应用。该工具提供开箱即用的构建工具配置，带来现代化的前端开发流程。只需几分钟即可创建并启动一个带热重载、保存时静态检查以及可用于生产环境的构建配置的项目.

主要功能:

1. 统一的目录结构
2. 本地调试
3. 热部署
4. 单元测试
5. 集成打包上线

安装

```
cnpm install vue-cli-g
#测试是否安装成功
#查看可以基于哪些模板创建vue应用程序，通常我们选择webpack
vue list
```

## 创建一个基于 webpack 模板的新项目

```
# 切换到指定的目录下，当然是你自己指定的
# 执行创建的命令
vue init webpack myvue
// npm init webpack myvue 错误
# 使用idea打开这个项目
# 安装需要的依赖
npm install
# package.json里启动
```

```
├── build/                      # webpack 编译任务配置文件：开发环境与生产环境
│   └── ...
├── config/
│   ├── index.js                # 项目核心配置
│   └── ...
├ ── node_module/                #项目中安装的依赖模块
   ── src/
│   ├── main.js                 # 程序入口文件
│   ├── App.vue                 # 程序入口vue组件
│   ├── components/             # 组件
│   │   └── ...
│   └── assets/                 # 资源文件夹，一般放一些静态资源文件
│       └── ...
├── static/                     # 纯静态资源（直接拷贝到dist/static/里面）
├── test/
│   └── unit/                   # 单元测试
│   │   ├── specs/              # 测试规范
│   │   ├── index.js            # 测试入口文件
│   │   └── karma.conf.js       # 测试运行配置文件
│   └── e2e/                    # 端到端测试
│   │   ├── specs/              # 测试规范
│   │   ├── custom-assertions/  # 端到端测试自定义断言
│   │   ├── runner.js           # 运行测试的脚本
│   │   └── nightwatch.conf.js  # 运行测试的配置文件
├── .babelrc                    # babel 配置文件
├── .editorconfig               # 编辑配置文件
├── .gitignore                  # 用来过滤一些版本控制的文件，比如node_modules文
件夹
├── index.html                  # index.html 入口模板文件
└── package.json                # 项目文件，记载着一些命令和依赖还有简要的项目描述
信息
└── README.md                   #介绍自己这个项目的，可参照github上star多的项
目。
build/
```

# Webpack

## 简介

> Webpack可以看做是**模块打包机**：它做的事情是，分析你的项目结构，找到JavaScript模块以及其它的一些浏览器不能直接运行的拓展语言（Scss，TypeScript等），并将其打包为合适的格式以供浏览器使用。

## 理解

> **就是把ES6规范的代码打包编译成ES5规范的代码来执行。**

## 安装

```
npm install webpack -g
# webpack客户端
npm install webpack-cli -g
```

## 验证

```
D:\code\personal\vue\webpack-study>webpack -v
webpack: 5.65.0
webpack-cli: 4.9.1
webpack-dev-server not installed
```

webpack.config.js:

```
entry                       # 入口文件，指定webpack使用哪个文件作为项目的入口
ouput                       # 输出，指定webpack把处理完成的文件放置到哪个路径
module                      # 模块，用于处理各种类型的文件
plugins                     # 插件，如：热更新、代码重用等
resolve                     # 设置路径指向
watch                       # 监听，用于设置文件改动后直接打包
```

## 实战

> 这个项目要实现用webpack将es6打包资源成es5,并在浏览器访问。

1. 创建一个项目

   就是在某个路径下创建一个文件夹，然后用idea打开即可。

2. 代码

   详见：https://gitee.com/liuwenxiu/webpack-study.git

3. 在terminal执行命令。

   ```
   webpack
   // 添加监听，会实时监听文件变化并重新打包
   webpack --watch
   ```

4. 访问index.html

# Vue Router路由

## 简介

> Vue.js 官方的路由管理器。
> 官网: https://router.vuejs.org/zh/

## 理解

> 用来转发请求的，根据访问路径跳转指定的组件。

## 安装

```
npm install vue-router --save-dev
```

## 实战

- 基础

```
1: 在唯一入口main.js中引入router
2: 创建router
    export default new VueRouter({
      routes: [
        {
          path: '/content',
          name: 'content',
          component: Content
        }
      ]
     })
3: 编写router里要引用的组件
4: App.vue中
    <router-link to="/shouye">首页</router-link>
    <router-view></router-view>

代码:
    https://gitee.com/liuwenxiu/myvue.git
```

- 进阶
  - 重定向

```
export default new VueRouter({
   routes: [{
       path: '/main',
       component: Main,
       children: [
         {
           path: '/user/list',
```

```
            component: UserList
          },{
             path: '/user/profile',
            component: UserProfile
          },{
             path: '', //进入main里默认走子组件: UserProfile
             redirect: '/user/profile'
          }
        ]
    },{
        path: '/login',
        component: Login
    },{
       path: '', // 重定向到/login。即访问http://localhost:8080/#/走http://localhost:8080/#/login
       redirect: '/login'
    }]
});
```

- 动态路由
  - 方式一

```
1: index.js
    ...
    {
        path: '/user/profile/:id',
        name: 'UserProfile',
        component: UserProfile
    }
    ...
2: Main.vue
    ...
     <router-link :to="{name:'UserProfile',params:{id:1}}">个人信息</router-link>
    ...
3:UserProfile.vue
    ...
    -- 必须套在div等Dom中才不会报错！！！
    <div>
        {{$route.params.id}}
    </div>
    ...
```

  - 方式二

```
1: index.js
    ...
    {
```

```
        path: '/user/profile/:id/:name',
        name: 'UserProfile',
        props: true,
        component: UserProfile
    }
    ...
2: Main.vue
    ...
     <router-link :to="{name:'UserProfile',params:{id:1,name:'刘文秀'}}">个人信息</router-link>
    ...
3:UserProfile.vue
    export default {
      name: "UserProfile",
      props:{
        id:'',
        name:''
      }
    }
    ...
    -- 必须套在div等Dom中才不会报错！！！
    <div>
        {{id}}<br>
        {{name}}
    </div>
    ...
```

- 路由模式

```
export default new VueRouter({
   mode: 'history', // 可以去掉路径中的#
   routes: [
   {
       path: '/login',
       component: Login
   },{
      path: '',
      redirect: '/login'
   }]
});
```

- 404页面

```
export default new VueRouter({
   mode: 'history', // 可以去掉路径中的#
   routes: [
   {
       path: '/login',
       component: Login
   },
   ...
   {
      path: '*', // 上面所有路径都没有匹配到
      component: NotFound
   }]
});
```

# Vue ElementUI工程

## 简介

> Element，一套为开发者、设计师和产品经理准备的基于 Vue 2.0 的桌面端组件库
> Vue.js 渐进式JavaScript 框架
> https://element.eleme.cn/#/zh-CN
> https://cn.vuejs.org/

## 理解

> vue最常与elementUI配合使用。
> 一个页面，一个js

## 安装

### 1.初始化webpack工程

```
vue init webpack vue-elementui
```

### 2.安装vue-router

```
npm install vue-router --save-dev
```

### 3.安装element-ui

```
npm i element-ui -S
```

### 4.安装依赖

```
npm install
```

### 5.安装SASS加载器(安装的是最新的node-sass)

```
npm install sass-loader node-sass --save-dev
```

### 6.启动测试

```
npm run dev
```

## 避坑指南

```
非常容易遇到node-sass版本问题:
    Module build failed: Error: Node Sass version 7.0.0 is incompatible with ^4.0.0.@node_sass
起因:
    node-sass与sass-loader版本号不对应导致。
分析:
    因为npm install sass-loader node-sass --save-dev安装的是最新的node-sass,可能会导致版本不一致。
解决:
 解决方案一:
    npm uninstall node-sass sass-loader //先删除//node-sass也是要删除的
    npm install sass-loader@版本号 node-sass@版本号  --save-dev//然后执行修改的对应版本号
    版本号可以覆盖 最好删除，减少空间占用
 解决方案二:
    node-sass 4.13.0
    sass-loader 7.3.1
    在package.json中修改这两个版本
注:
    更改后需要执行npm install。
```

附:

## 实战

> 代码路径: https://gitee.com/liuwenxiu/vue-elementui.git

```
1: 唯一入口main.js
    引入router
    引入elementui
    引入elementui css
2: router还是以前的写法
3: .vue文件
```

## 嵌套路由

```
export default new VueRouter({
   routes: [{
       path: '/main',
       component: Main,
       children: [  //嵌套路由
         {
           path: '/user/list',
           component: UserList
         },{
            path: '/user/profile',
           component: UserProfile
         }
       ]
   },{
       path: '/login',
       component: Login
   }]
});
```