



Vue基础

写代码很爽的一种体验



黑马程序员
www.itheima.com

传智教育旗下
高端IT教育品牌



目录

Contents

- ◆ Vue基本概念
- ◆ @vue/cli脚手架
- ◆ Vue指令

学习目标

Learning Objectives

1. 能够说出Vue的概念和作用
2. 能够使用@vue/cli脚手架工程化开发
3. 能够熟练Vue指令



目录

Contents

- ◆ Vue基本概念
- ◆ @vue/cli脚手架
- ◆ Vue和指令



Vue基本概念



问题1: 原生js – 代码复杂开发效率慢?

问题2: 有没有一种简单快速的方式?

1.0_为什么学习Vue

更少的时间, 干更多的活. 开发网站速度快



原生js



jQuery



Vue

- 原生js做法

```
<ul id="myUl"></ul>
<script>
  let arr = ["春天", "夏天", "秋天", "冬天"];
  let myUl = document.getElementById("myUl");
  for (let i = 0; i < arr.length; i++) {
    let theLi = document.createElement("li");
    theLi.innerHTML = arr[i];
    myUl.appendChild(theLi);
  }
</script>
```

```
</script>
}
```

ωλη j : gbbuqcu j q (cu67 j) !

- Vue做法

```
<li v-for="item in arr">{{item}}</li>
```

```
<script>
```

```
  new Vue({
```

```
    // ...
```

```
    data: {
```

```
      arr: ["春天", "夏天", "秋天", "冬天"]
```

```
    }
```

```
  })
```

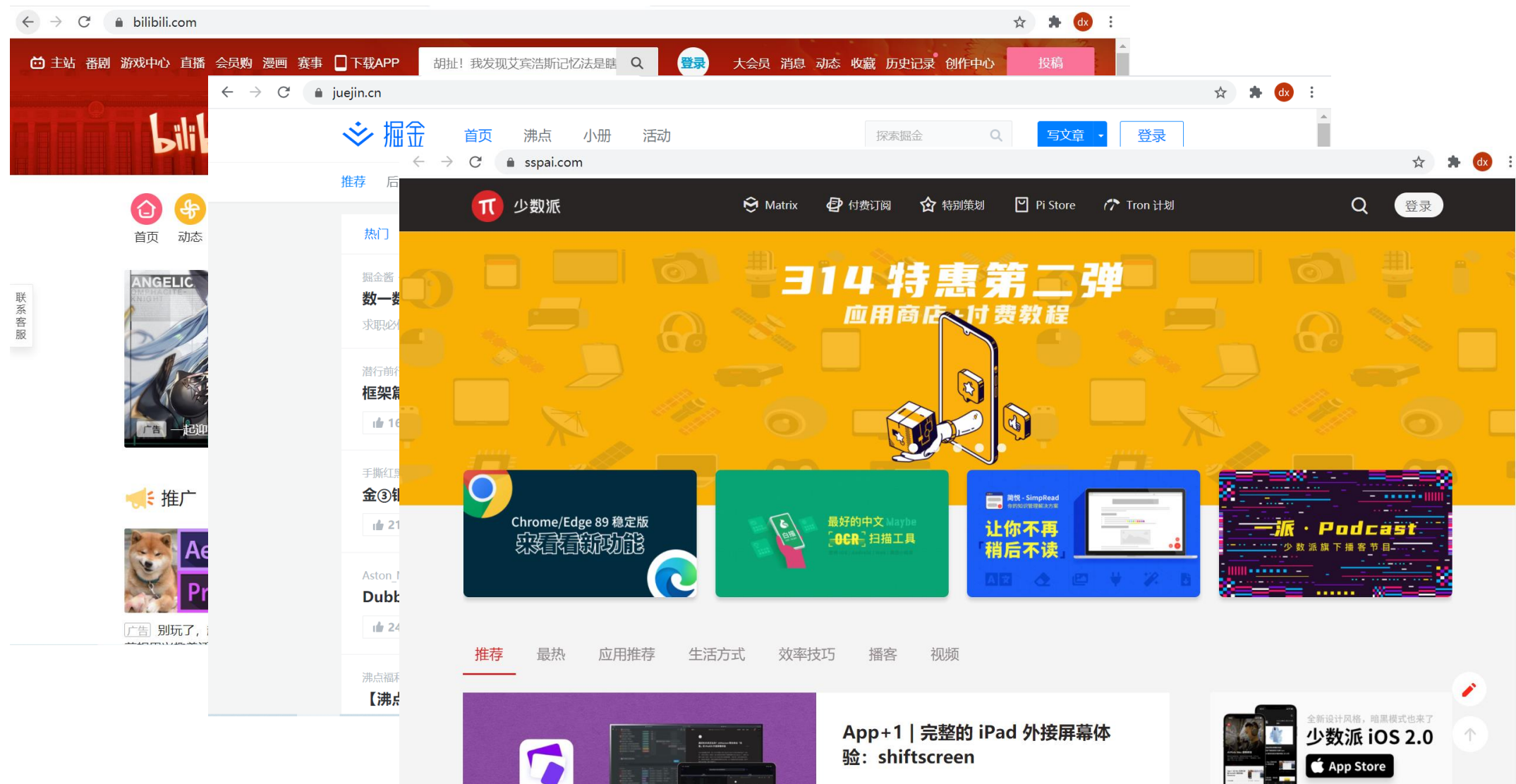
```
</script>
```

```
</script>
```

```
  })
```

1.0_为什么学习Vue

企业里都使用Vue开发网站



1.0_为什么学习Vue

市场上90%工作都要求会Vue, 会Vue拿高薪, 甚至JAVA或测试都要学点Vue

职位描述

前端开发工程师 (vue)

10-20k·12薪 年终奖

Web前端开发13-18k·12薪

应聘职位

微信扫码分享

收藏

岗位职责:

1. 负责项目前端
2. 持续优化前端
3. 解决前端开发

岗位要求:

1. 计算机或相关
2. 熟悉MVVM模
3. 熟悉HTML/H
4. 了解前端自动
5. 具备工程化的
6. 对用户体验、

职位描述:

岗位职责:

1. 基于vue框架的应用和
 2. 参与产品的前端开发,
 3. 提升产品的用户体验,
 4. 设计用户交互模式, 提
- 负责公司PC管理端, 小
2. 负责解决Web端和移动
 3. 负责研究前端技术发展
 4. 负责参与Web前端开发

岗位要求:

1. 全日制本科以上学历,
2. 熟练Web前端技术 (H
3. 至少熟练掌握一种主流
4. 熟练HTML5、CSS、JS
5. 熟练vue.js框架, 熟练
6. 具备较强的学习能力,
7. 开发过网站或者iOS/A
8. 熟悉金融行业项目者优

11k-19k·13薪 / 北京 / 经验

Vue

Node.js

2天前 发布于拉勾网

职位诱惑:

13薪, 项目奖金, 福利好. 带薪年假.

职位描述:

岗位职责:

- 1 主要负责业务部门的数据平台开发
 - 2 开发和完善数据产品的前端系统, 包
 3. 负责数据采集, 清洗, 存储监控等
- 招聘要求:
- 1 熟练掌握响应式Web开发框架 (A
 - 2 掌握 JavaScript 基本原理, 熟悉
 - 3 熟悉数据可视化库, 如ECharts/D

杭州

大专及以上 | 1-3年 | 语言不限 | 年龄不限

年终奖金

股票期权

领导好

发展空间大

扁平管理

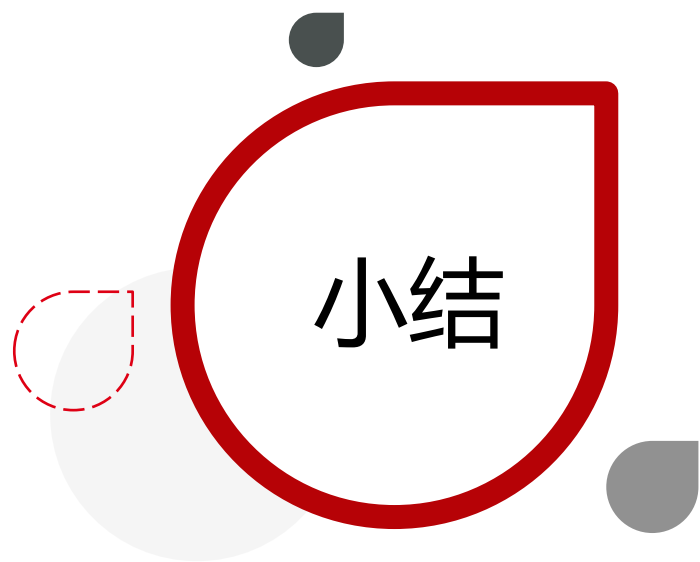
岗位晋升

带薪年假

优秀员工奖

职位描述:

- 1、大专以上学历、有2年以上Web前端开发经验
- 2、精通VueJS, 至少完整做过vue一个项目
- 3、精通 CSS3、CSS2/HTML5, 熟悉Less或Sass语法, 比较深刻理解W3C标准
- 4、熟悉HTTP协议, 连接TCP/IP的基本工作原理, 熟悉基本web标准规范
- 5、使用过webpack等工具构建和优化项目
- 6、关注新技术, 学习能力和独立解决问题的能力。有前端框架构建, 组件化经验优先
- 7、熟悉canvas优先
- 8、对前端技术和团队工作能有持续的热情, 逻辑思维清晰, 拥有主人翁心态, 善于与团队交流和合作, 勇于接受挑战并能够承受压力, 自我驱动, 不断学习和分享, 提高自身团队内外的影响力



我们为什么学习Vue

1. 开发更快速, 更高效
2. 企业开发都在使用
3. 前端工程师必备技能, 高薪



问题: Vue是什么?

1.1_Vue是什么

- 渐进式javascript框架, 一套拥有自己规则的语法
- 官网地址: <https://cn.vuejs.org/> (作者: 尤雨溪)





目标

1. 渐进式概念
2. 库和框架概念

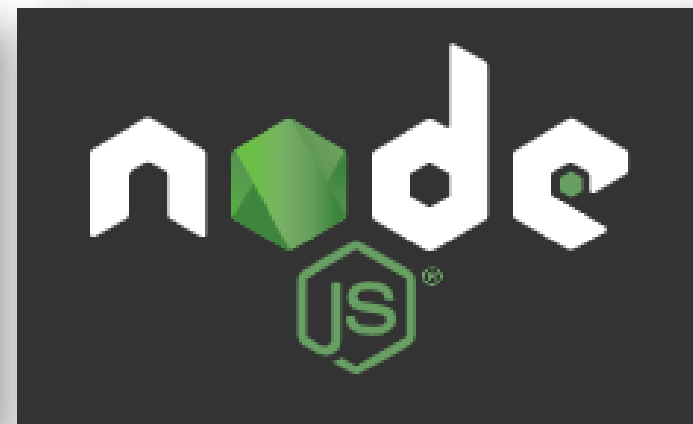
1.1_生活里_渐进式

- 渐进式: 逐渐进步, 想用什么就用什么, 不必全都使用



1.1_web里_渐进式

- HTML能写网页, CSS能让网页更好看, JS赋予网页交互效果, jQ写的更快, node可以提供动态数据



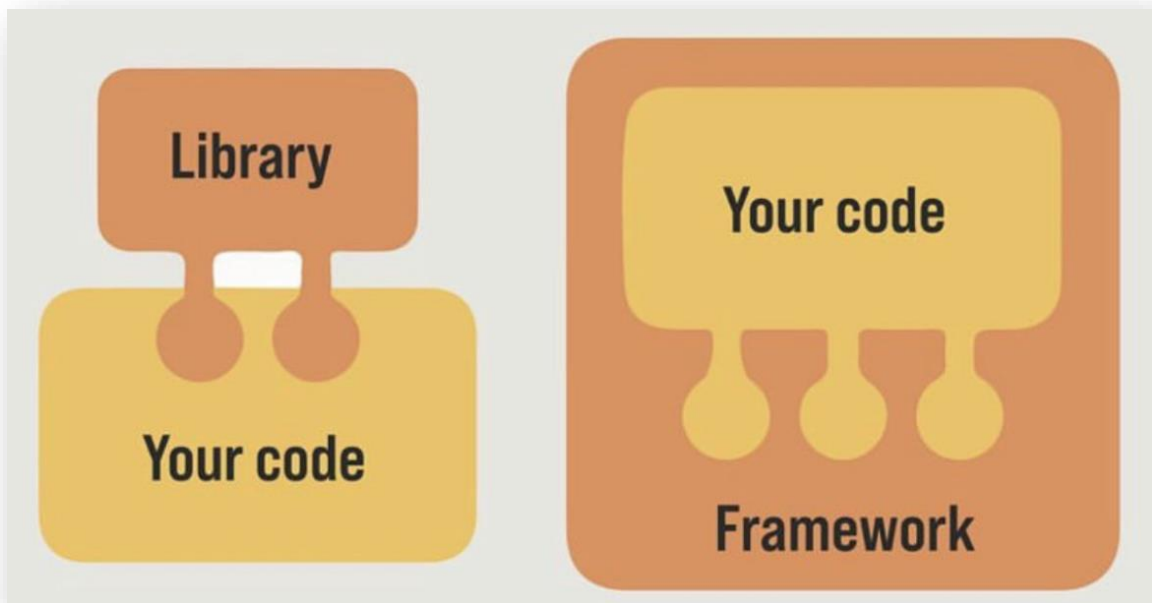
1.1_Vue里_渐进式

Vue从基础开始, 会循序渐进向前学习, 如下知识点可能你现在不明白, 但是学完整整个Vue回过头来看, 会很有帮助

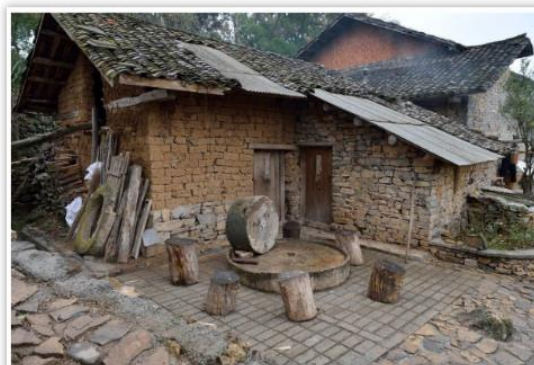


1.1_库和框架

- 库: 封装的属性或方法 (例jQuery)
- 框架: 拥有自己的规则和元素, 比库强大的多 (例Vue)



库



框架





Vue是什么?

Vue是一个javascript渐进式框架

什么是渐进式呢?

渐进式就是逐渐使用, 集成更多的功能

什么是库和框架呢?

库是方法的集合, 而框架是一套拥有自己规则的语法

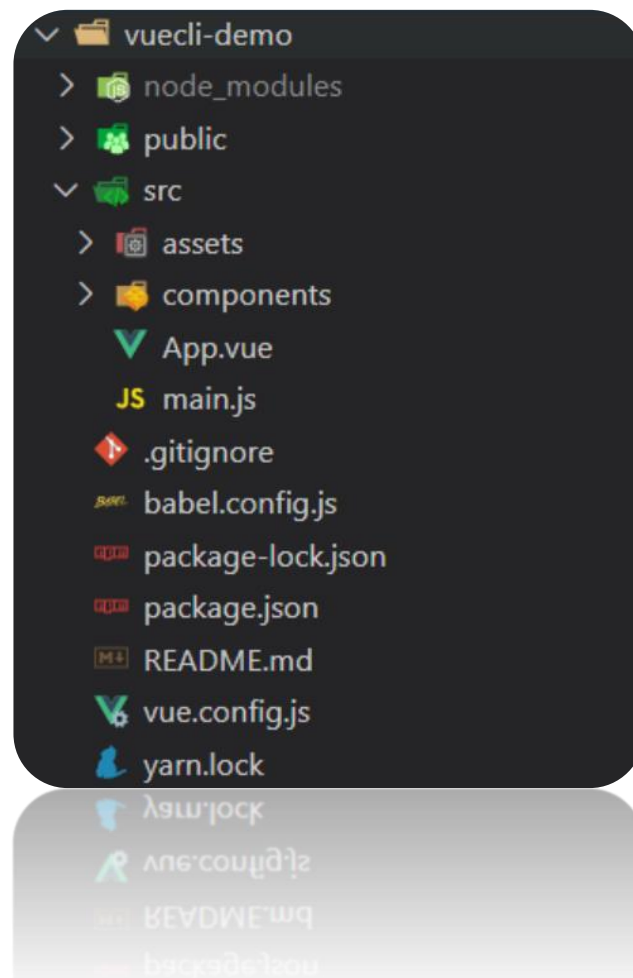
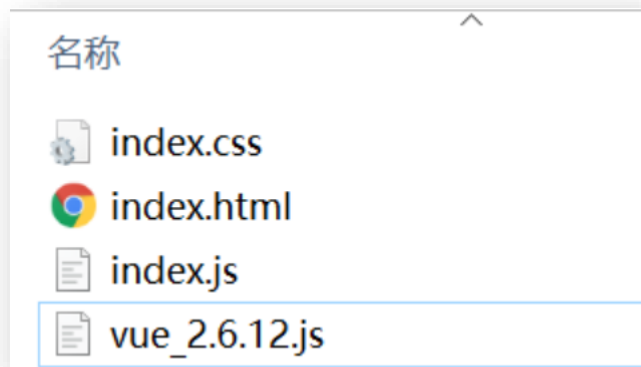


问题1: 我们在哪里去写Vue的代码呢?

问题2: 有没有好的学习方法, 事半功倍?

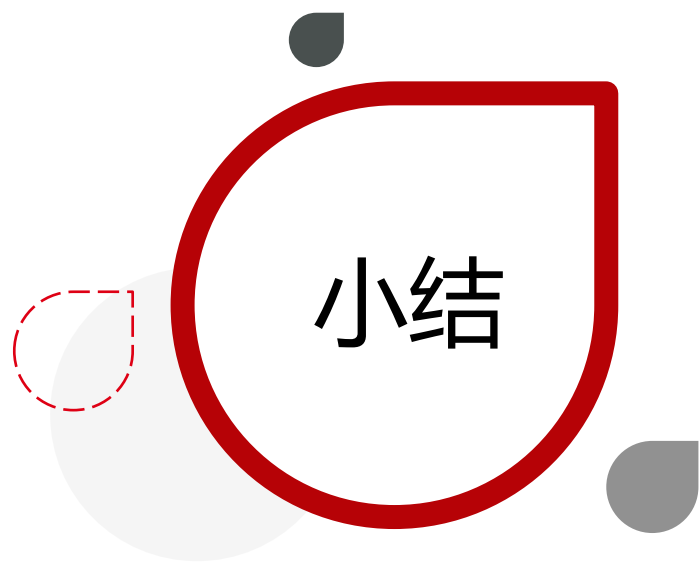
1.2_Vue学习方式

- 传统开发模式：基于html文件开发Vue
- 工程化开发方式：在webpack环境中开发Vue，这是最推荐, 企业常用的方式



1.2_Vue如何学

1. 知识点自测最好做到了如指掌 - 做不到只能花30分钟去记住结论和公式
2. 记住Vue指令作用, 基础语法 - 弄一个字典(一一映射关系)
3. 在课上例子, 练习, 案例, 作业, 项目中, 反复磨练使用
4. 学会查找问题的方式和解决方式(弄个报错总结.md, 避免反复进坑)



我们在哪里书写Vue的代码呢？

在工程化环境下用Vue开发项目

我们如何才能学好Vue呢？

多练多用多总结



目录

Contents

- ◆ Vue基本概念
- ◆ @vue/cli脚手架
- ◆ Vue指令



@vue/cli脚手架



思考

问题1: 新开发模式是不是离不开webpack?

问题2: 自己配置webpack是不是非常痛苦?

问题3: 有没有一套搭建好的, 拿来即用的环境呢?

2.0_@vue/cli和脚手架介绍

- @vue/cli是Vue官方提供的一个全局模块包(得到vue命令), 此包用于创建脚手架项目
- 脚手架是为了保证各施工过程顺利进行而搭设的工作平台



2.0_脚手架好处

- 开箱即用
- 0配置webpack
 - babel支持
 - css, less支持
 - 开发服务器支持



小结

问题1: 用Vue开发项目, 需要自己配置webpack吗?

Vue官方提供了@vue/cli包-可以创建脚手架项目

问题2: 使用脚手架好处是什么?

0配置开箱即用, 基于它开发我们的项目



思考

问题1: 以前如何安装一个全局包呢?

问题2: 如何用yarn安装@vue/cli这个全局包呢?

2.1_@vue/cli安装

- 全局安装@vue/cli 模块包

```
yarn global add @vue/cli
```

- 查看是否成功

C:\Windows\System32\cmd.exe

```
C:\Users>vue -V  
@vue/cli 4.5.12
```

```
C:\Users>
```

```
C:/Users>
```




小结

如何安装全局包

yarn global add 包名 / npm install 包名 -g

我们会得到什么呢

全局包会在计算机中配置全局命令(例: vue命令)



问题: 有了vue命令, 如何创建一套脚手架项目?

2. 选择模板和包管理器, 等待脚手架项目创建完毕

注意: 项目名不能有大写字母, 中文和特殊符号

```

? Generating README.md...
? Successfully created project vuecli-demo.
? Get started with the following commands:

$ cd vuecli-demo
$ yarn serve

C:\Users\lidongxu\Desktop\study>

```

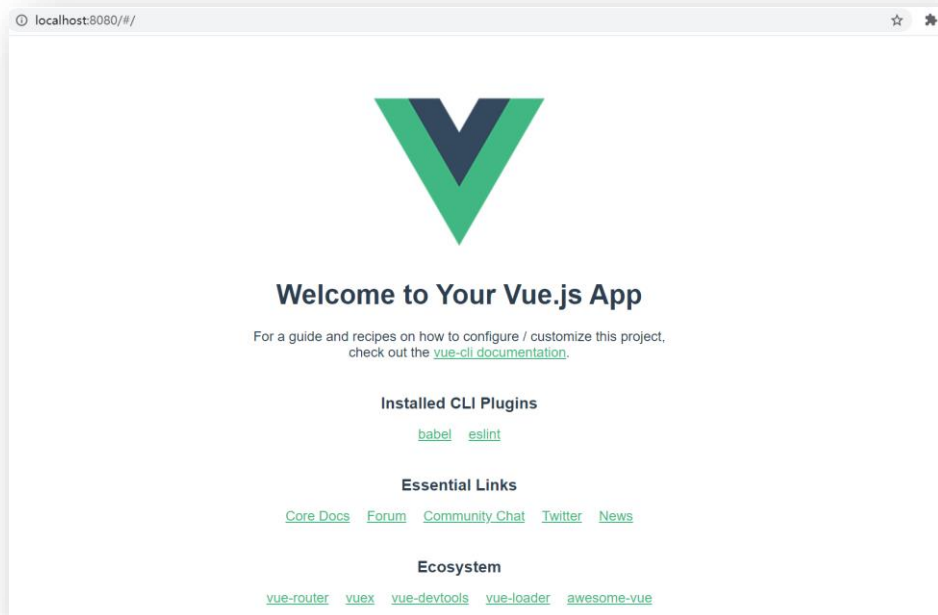
2.2_@vue/cli 启动开发服务

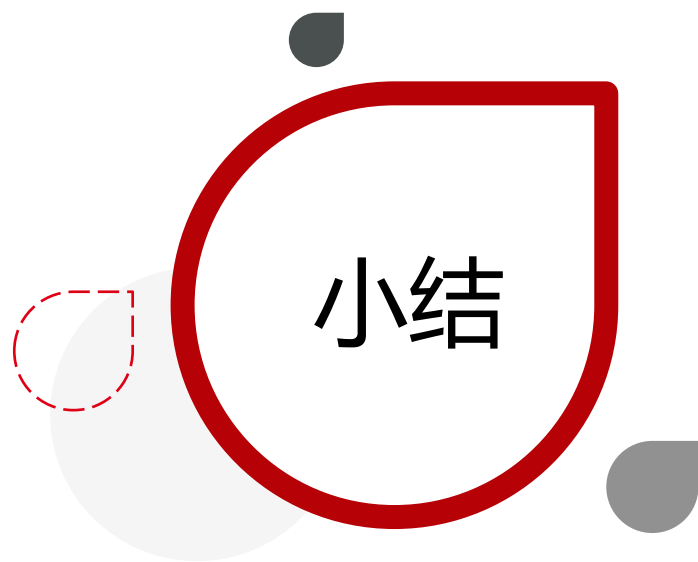
1. cd进入项目下, 启动内置的webpack本地热更新开发服务器 – 浏览项目页面

```
cd vuecli-demo
```

```
yarn serve
```

2. 如果未自动弹出浏览器, 手动打开输入提示的域名+端口浏览项目页面





1. 如何创建一个开箱即用的脚手架项目

`vue create` 项目名

2. 如何在网页上浏览这个项目

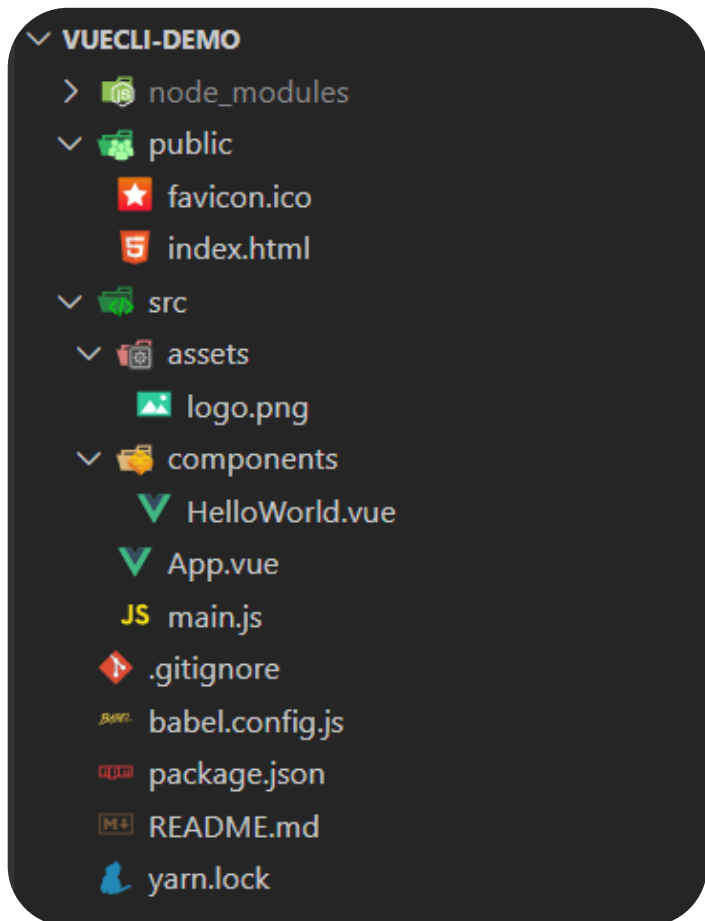
`yarn serve` 启动本地热更新开发服务器



问题: 脚手架项目里各个文件及代码都有什么作用呢?

2.3_@vue/cli 目录和代码分析

目标：文件夹和文件含义, 关键文件里代码意思





小结

脚手架里主要文件和作用？

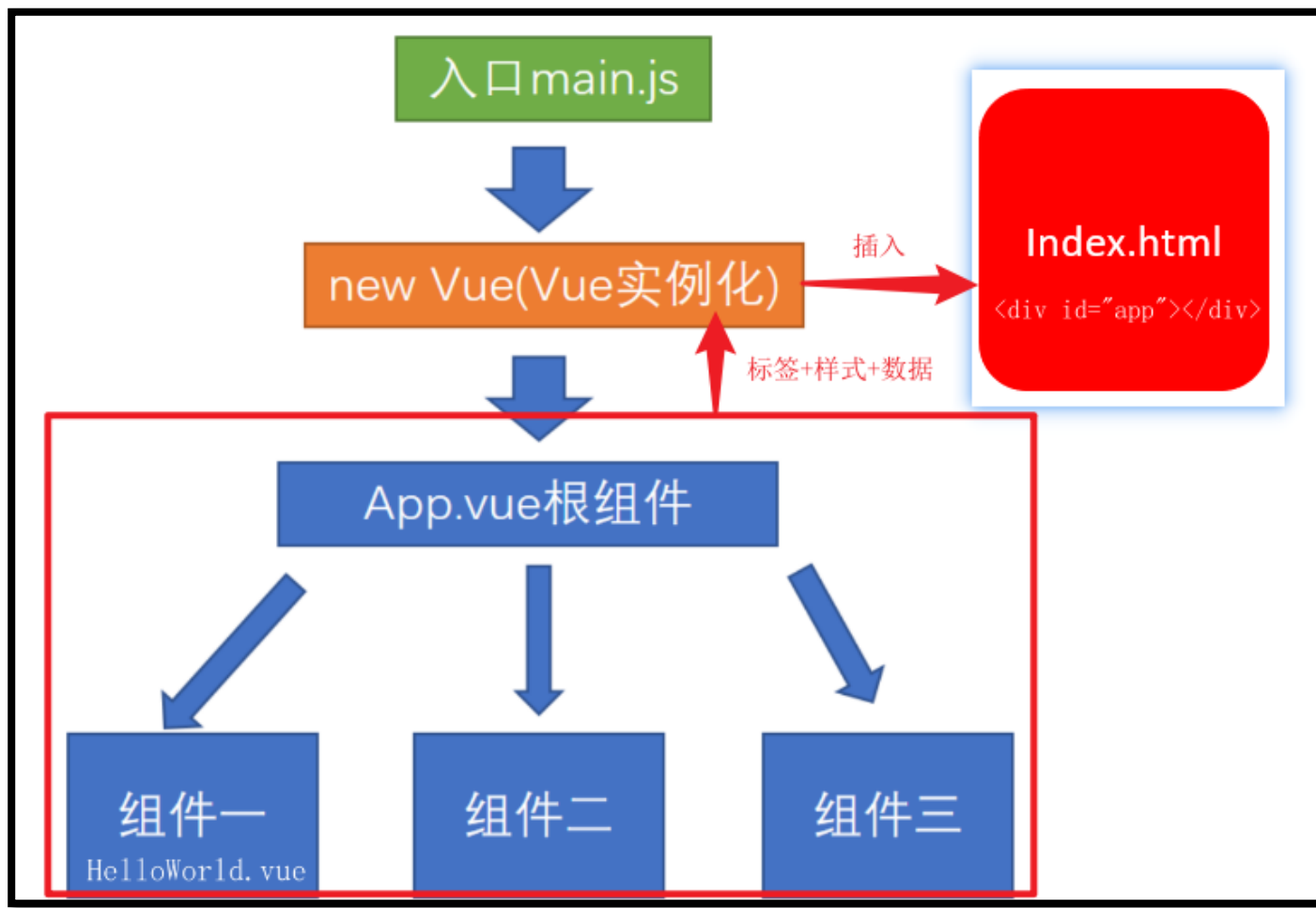
1. node_modules – 都是下载的第三方包
2. public/index.html – 浏览器运行的网页
3. src/main.js – webpack打包的入口
4. src/App.vue – Vue页面入口
5. package.json – 依赖包列表文件



问题: 项目入口和各文件关系是什么呢?

2.4_项目架构了解

目标：知道项目入口，以及代码执行顺序和引入关系





小结

main.js和App.vue, 以及index.html作用和关系?

1. main.js – 项目打包主入口 – Vue初始化
2. App.vue – Vue页面主入口
3. index.html – 浏览器运行的文件
4. App.vue => main.js => index.html



思考

问题1: 端口号我不想要8080, 比如要3000可以改吗?

问题2: 学习webpack中, 在哪个文件里修改端口号?

2.5_@vue/cli 自定义配置

目标：项目中没有webpack.config.js文件，因为Vue脚手架项目用的vue.config.js

src并列处新建vue.config.js, 填入配置, 重启webpack开发服务器

```
module.exports = {  
  ...  
  devServer: { // 自定义服务配置  
    port: 3000,  
    open: true  
  }  
}
```



脚手架项目配置文件是什么？

src并列处准备vue.config.js配置文件



思考

问题1: 以前写代码很随意, 有很多根本不使用的变量?

问题2: 项目开发完, 我们还得回去挨个清理?

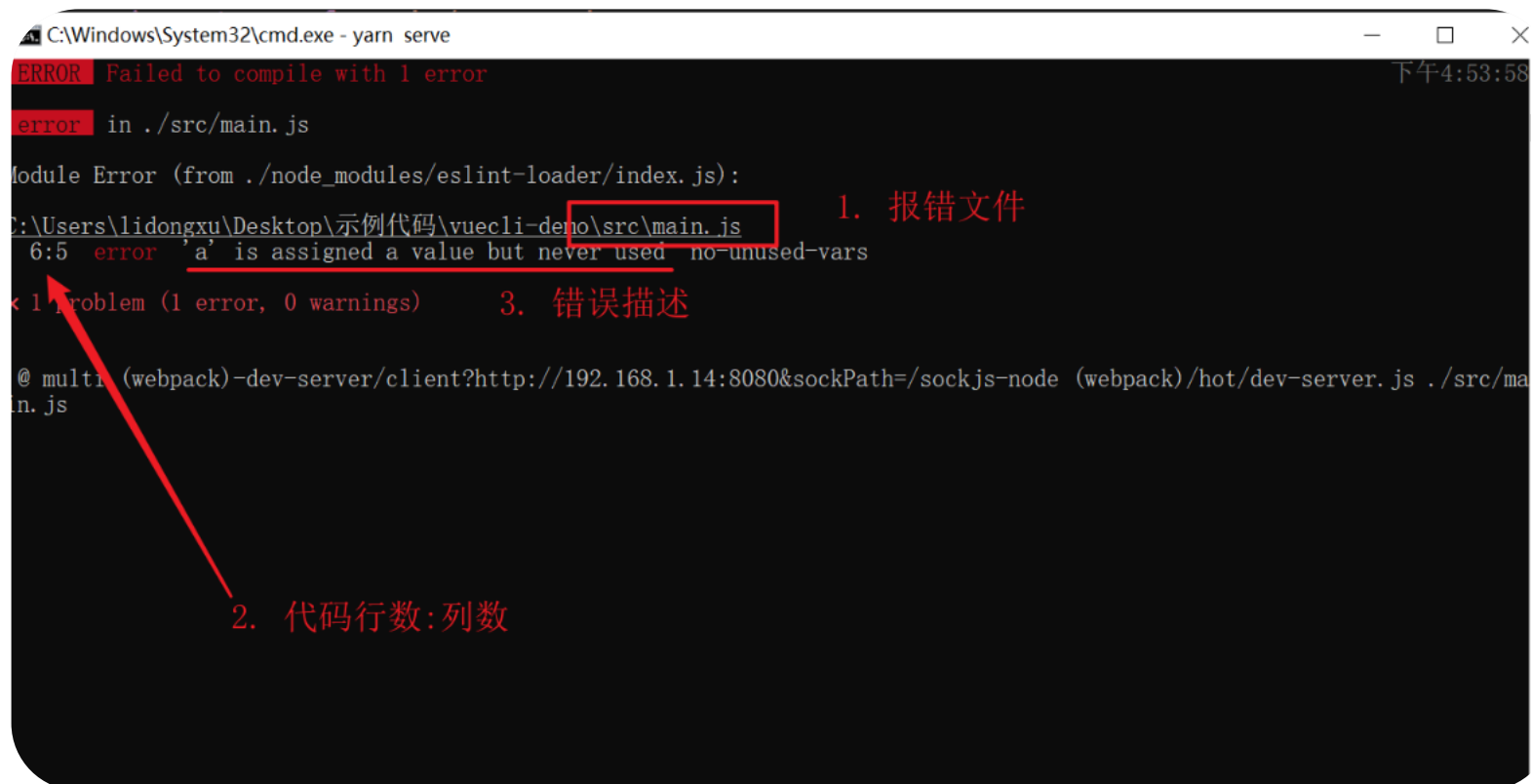
问题3: 有没有一种技术, 能够实时帮助我们检查代码呢?

2.6_eslint检查代码

目标: eslint是一种代码检查的工具

如果写代码违反了eslint的规则-报错

演示: 在main.js中随便定义变量 – 不使用 – 观察eslint报错



```
C:\Windows\System32\cmd.exe - yarn serve
ERROR Failed to compile with 1 error
error in ./src/main.js
Module Error (from ./node_modules/eslint-loader/index.js):
C:\Users\lidongxu\Desktop\示例代码\vuecli-demo\src\main.js
6:5 error 'a' is assigned a value but never used no-unused-vars
1 problem (1 error, 0 warnings)
@ multi (webpack)-dev-server/client?http://192.168.1.14:8080&sockPath=/sockjs-node (webpack)/hot/dev-server.js ./src/main.js
```

1. 报错文件

2. 代码行数:列数

3. 错误描述

2.6_处理eslint代码检查

方式1: 手动解决掉错误, 以后项目中会讲如何自动解决

方式2: 暂时关闭eslint检查(因为现在主要精力在学习Vue语法上), 在vue.config.js中配置后重启服务

```
module.exports = {  
  ...  
  // ...其他配置  
  lintOnSave: false // 关闭eslint检查  
}
```



小结

eslint是什么?

eslint是代码检查工具, 违反规定就报错

如何暂时关闭?

在vue.config.js中设置lintOnSave为false重启服务器即可

A decorative graphic on the left side of the slide. It features a large red-outlined hexagon in the center containing the Chinese characters '思考' (Thinking). Surrounding this central hexagon are several other hexagons: a light gray one at the top left, a dark red one at the top right, a dark gray one at the bottom left, and a red-outlined one at the bottom right. A dashed line connects the central hexagon to the dark gray one at the bottom left.

思考

问题1: 代码写在一起, 留意class类名, js变量名, 别重复?

问题2: 现在有没有办法避免这个问题呢?

2.7_单vue文件讲解

- Vue推荐采用.vue文件来开发项目
- template里只能有一个根标签
- vue文件-独立模块-作用域互不影响
- style配合scoped属性, 保证样式只针对当前template内标签生效
- vue文件配合webpack, 把他们打包起来插入到index.html

```
<!-- template必须, 只能有一个根标签, 影响渲染到页面的标签结构 -->
<template>
  <div>欢迎使用vue</div>
</template>

<!-- js相关 -->
<script>
export default {
  name: 'App'
}
</script>

<!-- 当前组件的样式, 设置scoped, 可以保证样式只对当前页面有效 -->
<style scoped>
</style>
```



单vue文件好处?

独立作用域, 不再担心变量重名问题

单vue文件使用注意事项?

要注意template里只能有一个根标签

A decorative graphic featuring a central red-outlined hexagon with the Chinese characters '思考' (Thinking) inside. Surrounding this central hexagon are several other hexagons: a light gray one at the top left, a solid dark red one at the top right, a dashed black one at the bottom left containing a small dark gray hexagon, and a red-outlined one at the bottom right.

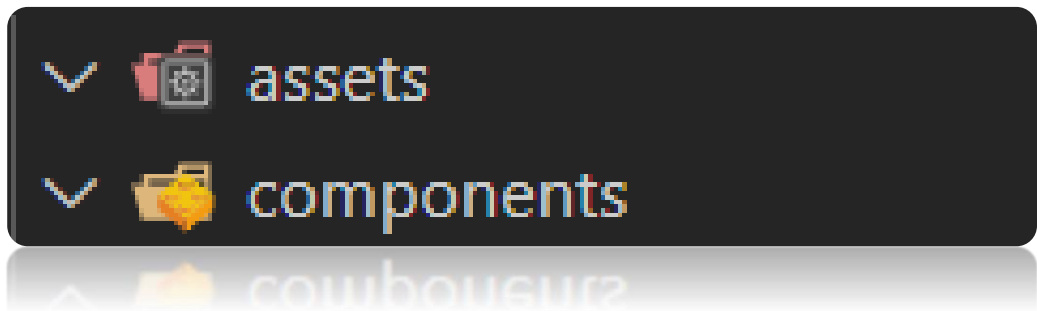
思考

问题1: 脚手架项目自带欢迎界面都是哪些文件和代码呢

问题2: 我们开发自己的需求, 需要他们吗?

2.8_脚手架项目_清理欢迎界面

- assets 和 components 文件夹下的一切都删除掉 (不要默认的欢迎页面)
- src/App.vue默认有很多内容, 可以全部删除留下template和script和style的框



```
<template>
  <div>

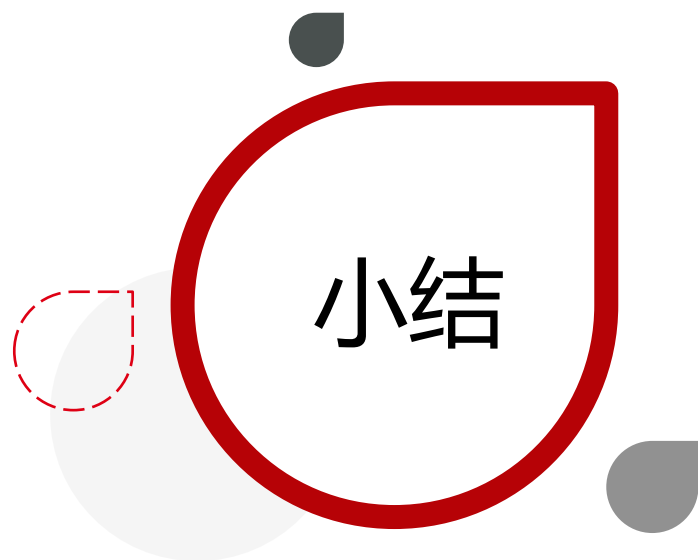
  </div>
</template>

<script>
export default {

}
</script>

<style scoped>

</style>
```



欢迎界面是哪些, 如何清除?

1. assets下的图片和components下的文件, App.vue初始代码
2. 删除即可, 但是要留下Vue项目入口页面App.vue文件



总结

1. @vue/cli是什么, 如何创建脚手架项目和启动?
2. 脚手架项目好处?
3. 脚手架项目入口都有哪些?
4. 单Vue文件的好处?



目录

Contents

- ◆ Vue基本概念
- ◆ @vue/cli脚手架
- ◆ **Vue指令**



Vue指令



问题1: 以前改变标签内容, 如何做?

问题2: 用Vue有没有更简单的方式?

3.0_Vue基础-插值表达式

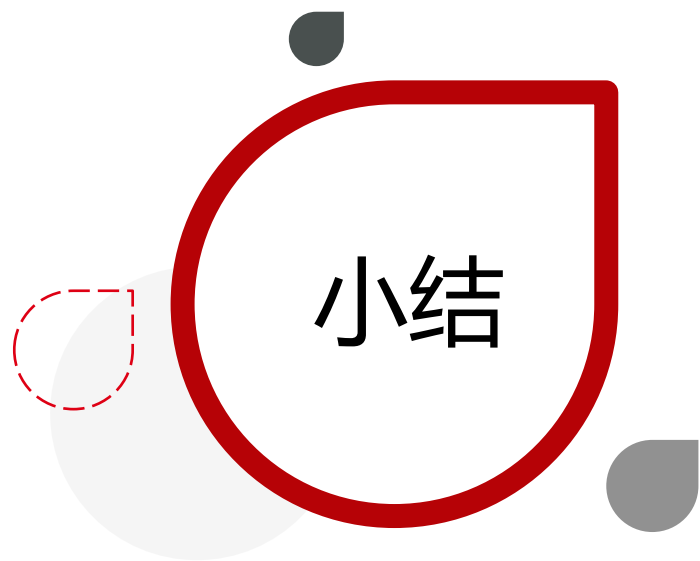
目标：在dom标签中, 直接插入vue数据变量

- 又叫: 声明式渲染/文本插值
- 语法: {{ 表达式 }}

- msg和obj是vue数据变量
- 要在js中data函数里声明

```
<template>
  <div>
    <h1>{{ msg }}</h1>
    <h2>{{ obj.name }}</h2>
    <h3>{{ obj.age > 18 ? '成年' : '未成年' }}</h3>
  </div>
</template>
```

```
<script>
export default {
  data() { // 格式固定, 定义vue数据之处
    return { // key相当于变量名
      msg: "hello, vue",
      obj: {
        name: "小vue",
        age: 5
      }
    }
  }
}
</script>
```



什么是插值表达式?

双大括号, 可以把vue数据变量直接显示在标签内

Vue中变量声明在哪里?

data函数返回的对象上, 用key属性声明



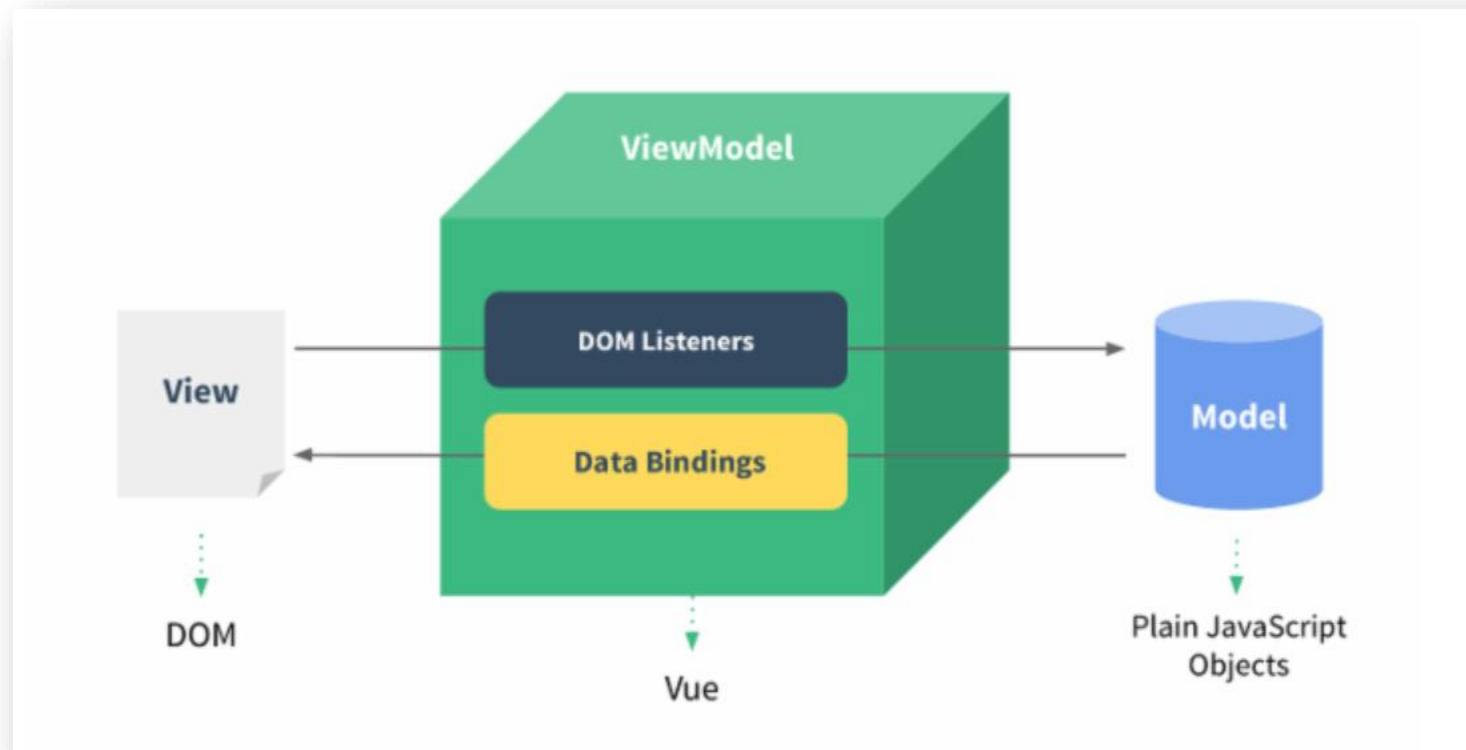
演示: 在上个代码基础上, 在devtool工具改变M层的变量, 观察V层(视图的自动同步)

问题: 为什么数据改变页面自动更新了?

3.1_vue基础-MVVM设计模式

目标：转变思维, 用数据驱动视图改变, 操作dom的事, vue源码内干了

- 设计模式: 是一套被反复使用的、多数人知晓的、经过分类编目的、代码设计经验的总结





什么是设计模式?

设计模式是对代码分层, 引入一种架构的概念

MVVM是什么?

MVVM(模型, 视图, 视图模型双向关联的一种设计模式)

MVVM好处?

减少DOM操作, 提高开发效率



学习目标

以v-开头的标签属性, 叫指令, 给标签添加额外功能

1. v-bind
2. v-on和修饰符
3. v-model和修饰符
4. v-text和v-html
5. v-show和v-if以及v-else
6. v-for



问题: 以前给a标签设置href如何做?

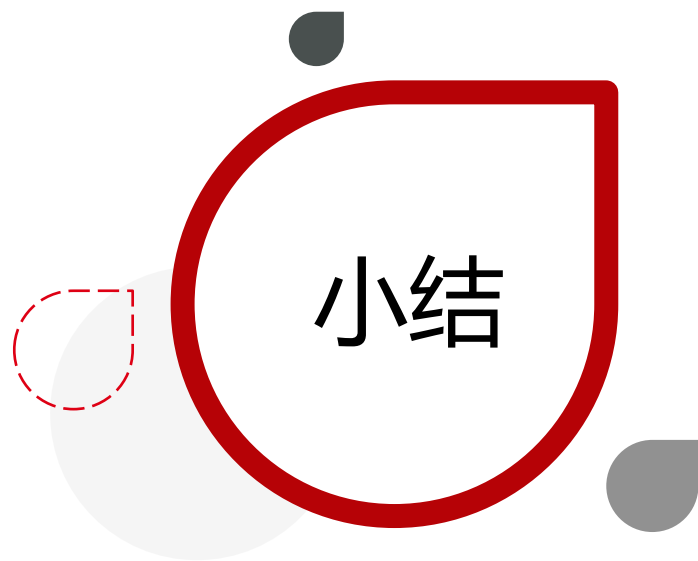
3.2_Vue指令-v-bind

目标：给标签属性设置Vue变量的值

- v-bind语法和简写
 - 语法：v-bind:属性名="vue变量"
 - 简写：:属性名="vue变量"

```
<!-- vue指令-v-bind属性动态赋值 -->
<a v-bind:href="url">我是a标签</a>


```



如何给dom标签的属性, 设置Vue数据变量?

:属性名= "Vue数据变量"



问题: 以前如何给按钮绑定点击事件呢?

3.3_Vue指令-v-on

目标：给标签绑定事件

- 语法
 - v-on:事件名 = “要执行的少量代码”
 - v-on:事件名 = “methods中的函数名”
 - v-on:事件名 = “methods中的函数名(实参)”
- 方法在methods选项定义

```
<p>你要买商品的数量: {{count}}</p>
<button v-on:click="count = count + 1">增加1</button>
<button v-on:click="addFn">增加1个</button>
<button v-on:click="addCountFn(5)">一次加5件</button>
<p v-on:click="addCountFn(2)">一次加2件</p>
```

```
<script>
  export default {
    // ...其他省略
    methods: {
      addFn(){ // this代表export default后面的组件对象(下属有data里return出来的属性)
        this.count++
      },
      addCountFn(num){
        this.count += num
      },
      subFn(){
        this.count--
      }
    }
  }
</script>
</script>
```

3.3_Vue指令-v-on

目标: v-on: 可以简写成@

- 语法:
 - @事件名 = "methods中的函数"

```
<button @click="subFn">减少</button>
```




小结

如何给dom标签绑定事件?

@事件名= "methods里的函数名"

如何给事件传值呢?

@事件名= "methods里的函数名(实参)"

A decorative graphic on the left side of the slide. It features a large red hexagon with a thick border in the center, containing the Chinese characters '思考' (Thinking). Surrounding this central hexagon are several other hexagons: a light gray one at the top left, a dark red one at the top right, a dark gray one at the bottom left, and a red outline one at the bottom right. A dashed line connects the central hexagon to the dark gray one at the bottom left.

思考

问题1: js原生如何阻止标签默认行为?

问题2: vue事件处理函数如何获取事件对象呢?

3.4_Vue指令-v-on事件对象

目标: Vue事件处理函数中, 拿到事件对象

- 语法
 - 无传参, 通过形参直接接收
 - 传参, 通过`$event`指代事件对象传给事件处理函数

```
<template>
  <div>
    <a @click="one" href="http://www.baidu.com">阻止百度</a>
    <hr>
    <a @click="two(10, $event)" href="http://www.baidu.com">阻止去百度</a>
  </div>
</template>
```

```
<script>
export default {
  methods: {
    one(e){
      e.preventDefault()
    },
    two(num, e){
      e.preventDefault()
    }
  }
}
</script>
```



小结

Vue事件处理函数, 如何拿到事件对象?

1. 无传参, 直接在形参接收
2. 有传参, 手动传入`$event`对象



思考

问题1: `e.preventDefault()`单词很长不好写吧?

问题2: 有没有一种更简单的方式实现呢?

3.5_Vue指令-v-on修饰符

目标：在事件后面.修饰符名 - 给事件带来更强大的功能

- 语法
 - @事件名.修饰符="methods里函数"
- 修饰符列表
 - .stop - 阻止事件冒泡
 - .prevent - 阻止默认行为
 - .once - 程序运行期间, 只触发一次事件处理函数

```
<template>
  <div @click="fatherFn">
    <!-- vue对事件进行了修饰符设置，在事件后面.修饰符名即可使用更多的功能 -->
    <button @click.stop="btn">.stop阻止事件冒泡</button>
    <a href="http://www.baidu.com" @click.prevent="btn">.prevent阻止默认行为</a>
    <button @click.once="btn">.once程序运行期间，只触发一次事件处理函数</button>
  </div>
</template>
</c6wbj9t6>
</qj^>
```



Vue有哪些主要修饰符, 都有什么功能?

1. .stop – 阻止事件冒泡
2. .prevent – 阻止默认行为
3. .once – 只执行一次事件处理函数

A decorative graphic featuring a central red-outlined hexagon with the word '思考' (Thinking) inside. Surrounding it are several other hexagons: a light gray one at the top left, a dark red one at the top right, a dark gray one at the bottom left, and a red-outlined one at the bottom right. A dashed line connects the central hexagon to the dark gray one at the bottom left.

思考

问题1: 以前我想判断用户是否按下回车怎么做?

问题2: 鼠标事件有修饰符, 那按键有修饰符吗?

3.6_Vue指令-v-on按键修饰符

目标：给键盘事件, 添加修饰符, 增强能力

- 语法:
 - @keyup.enter - 监测回车按键
 - @keyup.esc - 监测返回按键

```
<template>
  <div>
    <input type="text" @keydown.enter="enterFn">
    <hr>
    <input type="text" @keydown.esc="escFn">
  </div>
</template>
```

- 更多修饰符: <https://cn.vuejs.org/v2/guide/events.html>



小结

按键修饰符如何使用？

@键盘事件.按键修饰符= “methods里函数名”

有哪些主要按键修饰符？

1. .enter – 只要按下回车才能触发这个键盘事件函数
2. .esc – 只有按下取消键才能触发这个键盘事件函数

练习 翻转世界

需求： 点击按钮 - 把文字取反显示 - 再点击取反显示(回来了)

提示： 点击事件里, 把Vue变量值挨个字母取反赋予回来 (提示跟数组有关系)



练习

翻转世界

分析：先静后动

- ①：定义变量 message: 'HELLO, WORLD'
- ②：插值表达式赋予到dom上, 准备按钮和文字
- ③：按钮绑定点击事件和函数
- ④：对message值用split拆分, 返回数组
- ⑤：数组元素翻转用reverse方法
- ⑥：再把数组用join拼接成字符串赋予给message
- ⑦：因为Vue是MVVM设计模式, 数据驱动视图, 所以视图自动改变

Hello, World

逆转世界



点击翻转字符串你有什么收货?

1. 写需求要先静态标签, 再考虑动态效果, 找好第一步干什么
2. 记住方法的特点 – 可以自己总结字典和口诀
3. Vue是靠数据驱动视图, 只需要关心数据变化即可



思考

问题1: js中获取表单的值, 赋予给变量?

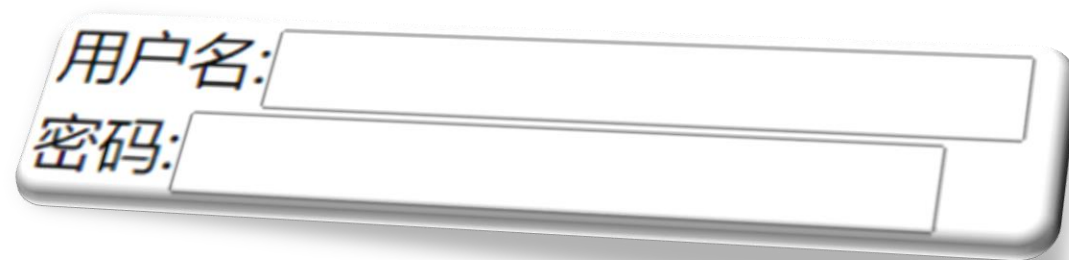
问题2: js变量改变, 同步显示到表单里?

问题3: Vue能否把他们互相关联起来呢?

3.8_Vue指令-v-model

目标：value属性和Vue数据变量, 双向绑定到一起

- 语法: `v-model="Vue数据变量"`
- 双向数据绑定
 - 变量变化 -> 视图自动同步
 - 视图变化 -> 变量自动同步
- 例子, 做个注册表单, 了解v-model在各种表单标签如何使用



用户名:

密码:



v-model用在哪里？

暂时只能用在表单标签上

v-model有什么作用？

把vue的数据变量和表单的value属性双向绑定在一起



问题: 下拉菜单, 复选框, 单选框如何绑定Vue变量呢?

3.8_Vue指令-v-model

目标: value属性和Vue数据变量, 双向绑定到一起

- 语法: v-model="Vue数据变量"
- 例子如下:



来自于: ▼

爱好 ☐ 抽烟 ☐ 喝酒 ☐ 写代码

性别 ☐ 男 ☐ 女

自我介绍



下拉菜单v-model写在哪里？

在select, value在option上

v-model用在复选框时, 需要注意什么？

v-model的vue变量是

非数组 – 关联的是checked属性

数组 – 关联的是value属性

vue变量初始值会不会影响表单的默认状态？

会影响, 因为双向数据绑定-互相影响



思考

问题1: 表单同步到变量里的 数 是什么类型的呢?

问题2: 能否让v-model转成数值再赋予给变量呢?

3.9_Vue指令-v-model修饰符

目标：让v-model拥有更强大的功能

- 语法: v-model.修饰符="Vue数据变量"
 - .number 以parseFloat转成数字类型
 - .trim 去除首尾空白字符
 - .lazy 在change时触发而非input时

```
<div>
  <span>年龄:</span>
  <input type="text" v-model.number="age">
</div>
<div>
  <span>人生格言:</span>
  <input type="text" v-model.trim="motto">
</div>
<div>
  <span>自我介绍:</span>
  <textarea v-model.lazy="intro"></textarea>
</div>
```

```
<\q!v>
<textarea v-model.lazy="intro"></textarea>
<span>自我介绍:</span>
```



小结

v-model有哪些修饰符, 提高我们编程效率?

1. .number – 转成数值类型赋予给Vue数据变量
2. .trim – 去除左右两边空格后把值赋予给Vue数据变量
3. .lazy – 等表单失去焦点, 才把值赋予给Vue数据变量



思考

问题1: JS原生的innerText和innerHTML作用和区别?

问题2: Vue有没有一种快速设置方式?

3.10_Vue指令-v-text和v-html

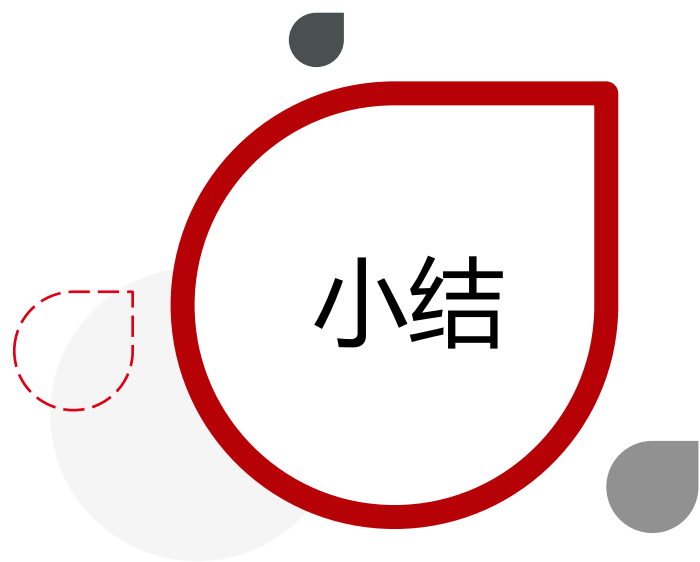
目标：更新DOM对象的innerText/innerHTML

- 语法：
 - v-text="Vue数据变量"
 - v-html="Vue数据变量"
- 注意：会覆盖插值表达式

```
<template>
  <div>
    <p v-text="str"></p>
    <p v-html="str"></p>
  </div>
</template>
```

```
<\f6wb]9f6>
```

```
<\qj^>
```

v-text和v-html有什么作用?

都可以设置标签显示的内容

区别是什么?

v-text把值当成普通字符串显示

v-html把值当成标签进行解析显示



问题1: JS原生/CSS如何隐藏一个标签?

问题2: 如何用一个变量来控制标签显示/隐藏呢?

3.11_Vue指令-v-show和v-if

目标：控制标签的隐藏或出现

- 语法:
 - v-show="Vue变量"
 - v-if="Vue变量"
- 原理
 - v-show 用的display:none隐藏 (频繁切换使用)
 - v-if 直接从DOM树上移除
- 高级
 - v-else的使用

```
<template>
  <div>
    <h1 v-show="isOk">v-show的盒子</h1>
    <h1 v-if="isOk">v-if的盒子</h1>

    <div>
      <p v-if="age > 18">我成年了</p>
      <p v-else>还得多吃饭</p>
    </div>
  </div>
```



vue如何控制标签显示/隐藏

v-show或v-if, 给变量赋予true/false, 显示/隐藏

区别是什么?

v-show是用css方式隐藏标签

v-if直接从DOM树上移除

v-if 可以配合v-else或者v-else-if使用

3.12_案例_折叠面板

案例 折叠面板

需求：点击展开或收起时，把内容区域显示或者隐藏

效果演示：



去md笔记里, 复制需要的标签和样式

案例 折叠面板

分析:

- ①: 准备标签和样式, 下载less模块和less-loader@5.0.0模块
- ②: 熟悉下标签结构和样式, 对应页面哪一部分
- ③: 按钮绑定点击事件
- ④: 声明变量isShow来控制下面标签是否显示/隐藏
- ④: 点击时, 改变isShow的值, 来影响页面效果

案例：折叠面板

芙蓉楼送辛渐

收起

寒雨连江夜入吴，
平明送客楚山孤。
洛阳亲友如相问，
一片冰心在玉壶。



折叠面板-案例思路是?

1. 还是先静态标签
2. 显示/隐藏用v-show和vue变量来控制(默认true)
3. 点击切换vue变量来达到目的



思考

问题1: JS原生把一个数组里学生姓名渲染到li上?

问题2: vue使用2行代码能搞定吗?

3.13_Vue指令-v-for

目标：列表渲染, 所在标签结构, 按照数据数量, 循环生成

- 语法:
 - v-for="(值变量, 索引变量) in 目标结构"
 - v-for="值变量 in 目标结构"
- 目标结构:
 - 可以遍历数组 / 对象 / 数字 / 字符串 (可遍历结构)
- 注意:
 - v-for的临时变量名不能用到v-for范围外

```


```



小结

v-for如何循环列表?

1. 谁想循环就把v-for写谁身上
2. v-for= “(值变量, 索引变量) in 目标结构” -一定注意in两边必须有空格
3. 可以遍历数组 / 对象 / 固定数字

v-for注意事项?

值变量和索引变量不能用到v-for范围以外



思考

1. Vue是什么, 为什么快?
2. @vue/cli和脚手架是什么?
3. Vue基础API(配置项data等)作用?
4. MVVM设计模式好处, 双向数据绑定是什么?
5. 插值表达式作用?
6. Vue指令和修饰符的作用? 你都学了哪些?



今日
作业

在md文档最后, 还有一些附加练习, 练习后再挑战作业



今日作业

1. 把课上例子, 练习, 案例做一遍
2. 完成<逛水果店>开发 – 具体看md文档
3. 完成<选择喜欢>开发 – 具体看md文档



传智教育旗下高端IT教育品牌