

基础概念 + vuecli + 指令上







- 1. 了解vue是什么
- 2. 知道vue-cli工具使用,能够创建一个vue项目
- 3. 掌握vue的基础指令,能够完成页面基本交互





- ◆ vue基本概念
- ◆ @vue/cli 脚手架的使用
- ◆ vue插值表达式
- ◆ vue指令
- ◆ 综合案例 文章标题编辑



# 案例效果:

标题 测试 编辑

标题: 如何成为一名前端老鸟?

频道: 前端

取消

确认





- ◆ vue基本概念
- ◆ @vue/cli 脚手架的使用
- ◆ vue插值表达式
- ◆ vue指令
- ◆ 综合案例 文章标题编辑





为什么要学习 vue 呢?

不能用原生 js开发么?



# 为什么要学习 vue

学习vue: 更少的时间,干更多的活。开发网站速度更快!



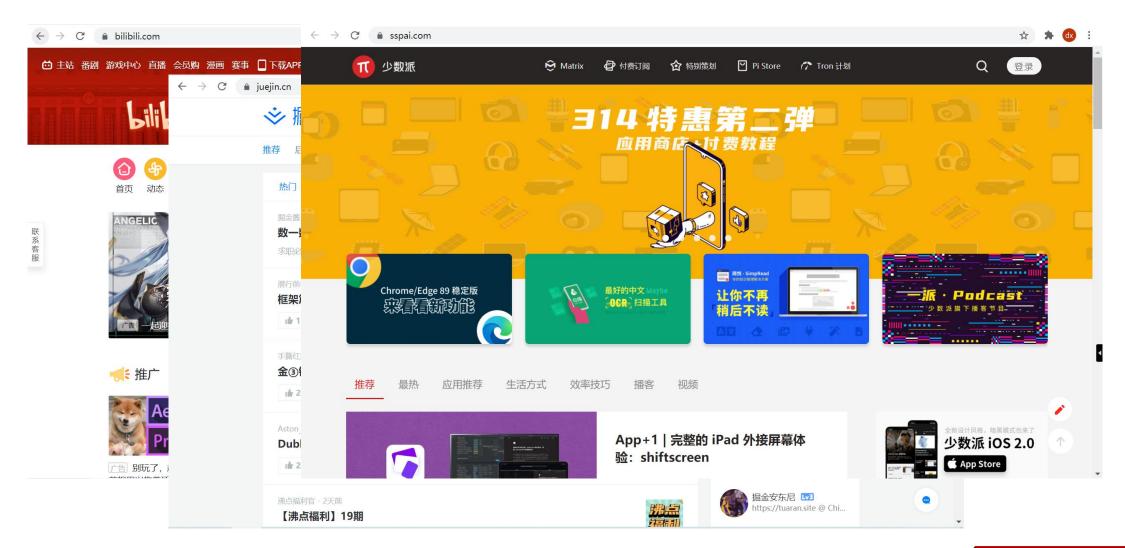
原生js



Vue



# 企业里都使用vue开发网站





# 市场上前端工作,基本都要求会vue,会vue可以拿高薪

职位描述	前端开发工程师(vue/angular) 浙江国技互联信息技术有  10-20k-12薪 年终奖金 绩效奖金 五脸一金	13-18k·12薪	应聘职位
岗位职责:		<b>₩</b> 17 1711	微信扫码分享 收藏
1. 负责项目前端	取位描述: 岗位职责:	大专及以上   1-3年   语言不限   年龄不限	
2. 持续优化前端	1.基于vue框架的应用和项目开发; 2.参与产品的前端开发,与后端工程师协作;	年终奖金 股票期权 领导好 发展空间大 扁平管理 岗位晋升	
3. 解决前端开发	<ul><li>3.提升产品的用户体验、前端性能;</li><li>4.设计用户交互模式,提升产品界面易用性和人性化;</li></ul>	带薪年假 优秀员工奖	
岗位要求:	负责公司PC管理端,小程序和APP端H5部分的页面开发,负责接口调证 2.负责解决Web端和移动端兼容性问题,优化前端体验。	取位描述:	
1. 计算机或相关	<ol> <li>负责研究前端技术发展,更新及提升技术。</li> <li>负责参与Web前端开发规范的制定及开发流程的优化。</li> </ol>	1、大专以上学历、有2年以上Web前端开发经验	
2. 熟悉MVVM核	岗位要求: 1.全日制本科以上学历,计算机相关专业,具有2年以上Vue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	
3. 熟悉HTML/H	2.熟练Web前端技术(HTML/CSS/Javascript等)的开发; 3.至少熟练掌握一种主流开发框架(react/vue/angular);	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	
4. 了解前端自动	4.熟练HTML5、CSS、JS、Ajax等前端开发技术,了解Java语言; 5.熟练vue.js框架,熟练掌握webpack,gulp等打包工具;	6、关注新技术,学习能力和独立解决问题的能力强。有前端框架构建,组件化经验优先 7、熟悉canvas优先	
5. 具备工程化的	6.具备较强的学习能力,基本功扎实; 7.开发过网站或者iOS/Android APP者优先;	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
6. 对用户体验、	8.熟悉金融行业项目者优先。	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	



我们为什么要学vue呢?

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



### vue是什么

vue 是一个 渐进式的 javascript 框架!

官网地址: https://cn.vuejs.org/ (作者: 尤雨溪)

渐进式?

框架?



渐进式 JavaScript 框架



起步



# 渐进式的理解

渐进式:逐渐增强, vue不强求你一次性在网站中运用学习所有的语法, 可以学一点用一点





# 库和框架的理解

1. 库:本质上是一些方法的集合。每次调用方法,实现一个特定的功能。 工具箱

如: moment axios

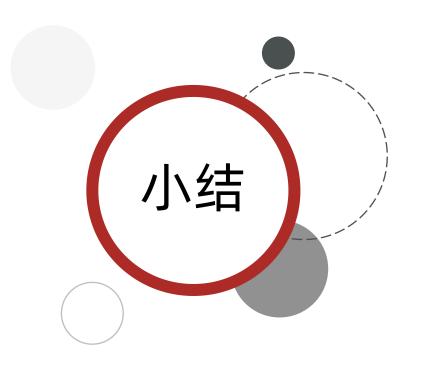
2. 框架: 是一套完整的解决方案。框架实现了大部分的功能, 我们需要按照框架的规则写代码。

如: vue react angular ... (虽然需要遵循规则,但是带来了高效)









vue是什么?

一个渐进式的javascript框架

什么是渐进式呢?

逐渐增强, 通过逐步学习, 集成更多的功能

什么是库和框架呢?

库是方法的集合

框架是一整套完整的项目解决方案



### Vue是一个MVVM的框架 (MVVM: 一种软件架构模式)

M: model数据模型(ajax获取到的数据) MVVM通过 数据双向绑定 让数据自动地双向同步

V: view视图(页面) V(修改视图) -> M(数据自动同步)

VM: ViewModel 视图模型 (操作视图+模型) M(修改数据) -> V(视图自动同步)



之前的思想,原生 dom驱动。无论修改什么页面内容,先找对象,操作dom。

现在的思想, vue 数据驱动。想更新视图,直接操作数据即可。数据变化,视图自动更新





#### 什么是 MVVM?

一种软件架构模式,决定了写代码的方式。

M、V、VM 分别是什么?

M: model 数据模型

V: view 视图

VM: viewModel 视图模型

原生和 vue 修改视图的思想分别是什么?

原生: dom驱动,修改视图,操作dom

vue:数据驱动,修改数据,操作数据即可

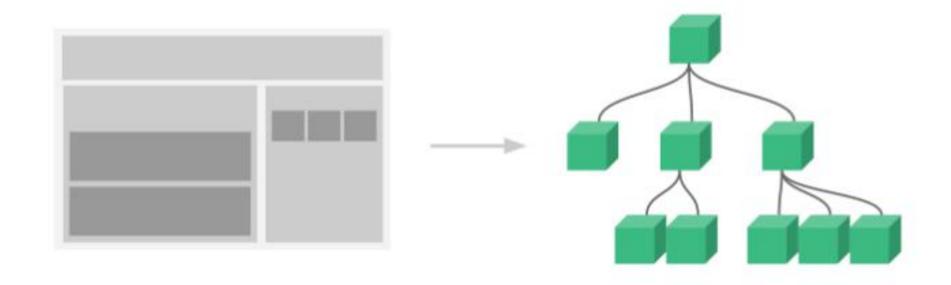


# vue组件化思想

组件化:一个组件会包含(HTML+CSS+JS),完整的页面可以拆分成多个组件。

#### 组件化的优点:

- 1. 容易维护
- 2. 便于复用(HTML+CSS+JS)







1. 什么是组件化?

将一个页面,拆分成一个个小组件的过程,就是组件化。

每个组件包含自己的 html+css+js

2. 组件化的好处是什么?

便于维护,便于复用



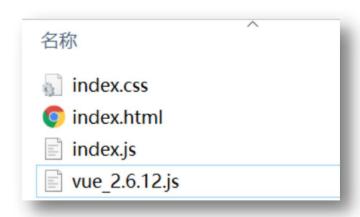


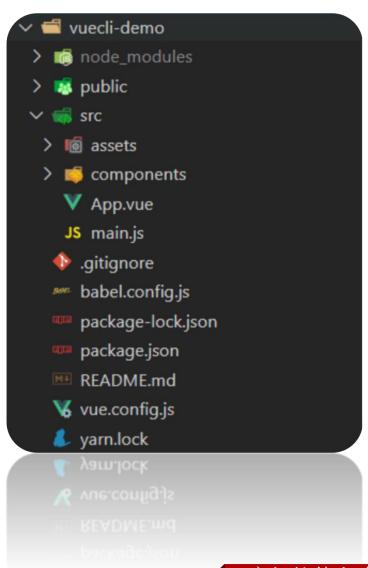
- ◆ vue基本概念
- ◆ @vue/cli 脚手架的使用
- ◆ vue插值表达式
- ◆ vue指令
- ◆ 综合案例 文章标题编辑



# 开发vue有两种方式:

- 1. 传统开发模式:基于html/css/js文件开发 vue
- 2. 工程化开发方式: 在webpack环境中开发vue。(推荐, 学习+项目)







注意:新开发模式基于 webpack!

问题1: 自己配置webpack是不是非常痛苦?

问题2: 有没有一套搭建好的, 拿来即用的环境呢?



### vue脚手架:

@vue/cli 也叫 vue脚手架, @vue/cli 是vue官方提供的一个全局命令工具这个命令可以帮助我们快速的创建一个vue项目的基础架子。

脚手架: 为了保证各施工过程顺利进行而搭设的工作平台



#### vue脚手架的好处:

- 1. 开箱即用
- 2. 零配置(不用你配webpack)
- 3. 内置babel等工具





### 什么是vue脚手架呢?

vue官方提供的一个全局命令工具。

可以 快速创建一个vue项目 的基础架子



### vue脚手架的基本使用:

1. 全局安装:

npm i @vue/cli -g 或 yarn global add @vue/cli

2. 查看vue版本:

vue --version

3. 找个目录,初始化一个vue项目:

vue create 项目名(不能用中文)

```
Please pick a preset: (Use arrow keys)

Default ([Vue 2] babel, eslint)

Default (vue 3) ([vue 3] babel, eslint)

Manually select features
```

4. cd 进入目录,启动项目,打包项目:

启动项目: yarn serve 或 npm run serve

打包项目: yarn build 或 npm run build

```
Pick the package manager to use when installing Use Yarn
Use NFM
```



我们说 @vue/cli 脚手架集成了 webpack,

但是我们怎么没有看到 webpack.config.js 呢?

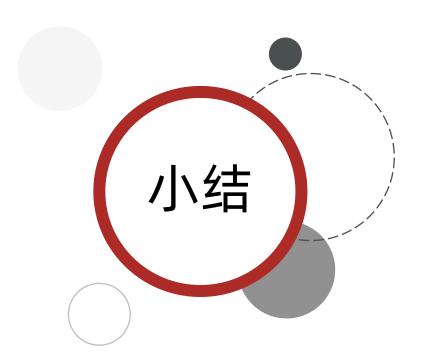


# 如何覆盖脚手架下的webpack配置:

- 1. 注意:我们在项目无法找到webpack.config.js文件,因为vue把它隐藏了。
- 2. 如果需要覆盖webpack的配置,可以修改vue.config.js文件,覆盖webpack配置

# vue.config.js





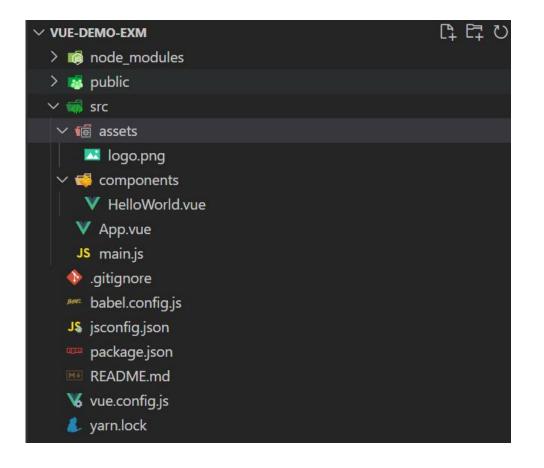
在vue脚手架中如何覆盖默认的webpack配置呢?

修改 vue.config.js 进行配置



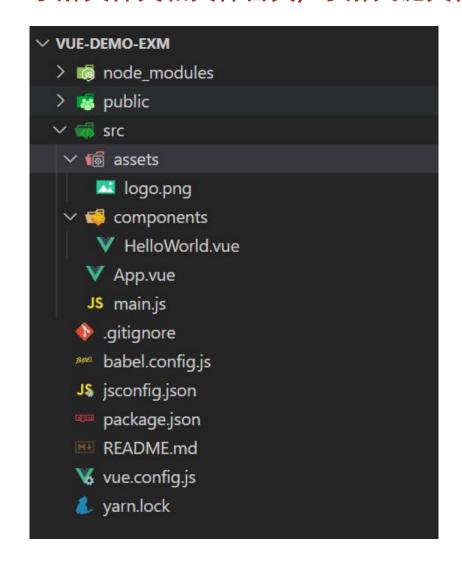


问题: 脚手架项目里各个文件及代码都有什么作用呢?





# 了解文件夹和文件含义,了解关键文件代码意思

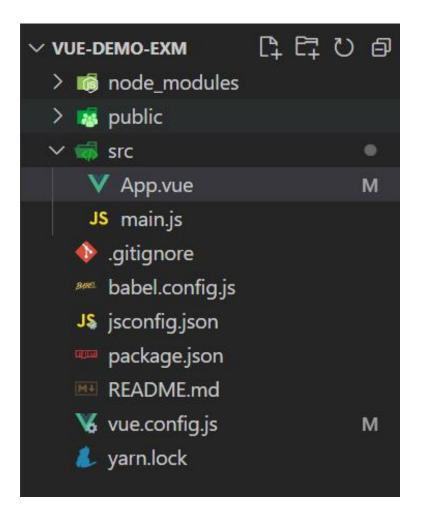


```
vuecil-demo
               # 项目目录
  ├─ node_modules # 项目依赖的第三方包
  ├─ public
               # 静态文件目录
    ├─ favicon.ico# 浏览器小图标
    └─ index.html # 单页面的html文件(网页浏览的是它)
         # 业务文件夹
   — src
    ── assets # 静态资源
     └─ logo.png # vue的logo图片
    ├─ components # 组件目录
     └─ HelloWorld.vue # 欢迎页面vue代码文件
    ├─ App. vue # 整个应用的根组件
    └─ main.js # 入口js文件
    — .gitignore  # git提交忽略配置
    — babel.config.js # babel配置
   — package.json # 依赖包列表
  ├── README.md # 项目说明
  └─ yarn.lock # 项目包版本锁定和缓存地址
```



### 目录分析与清理:

- 1. public/index.html不用动,提供一个最基础的页面
- 2. src/main.js不用动, 渲染了App.vue组件
- 3. src/App.vue默认有很多的内容,可以全部删除
- 4. assets 文件夹 和 components 直接删除





#### 脚手架里主要文件和作用?

- 1. node\_modules 下载的第三方包
- 2. public/index.html 运行的浏览器网页
- 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





.vue 文件是什么文件呢? 以前怎么没见过呢。



### 单文件组件: 一个 .vue 文件就是一个组件

1. 安装插件 (语法高亮)



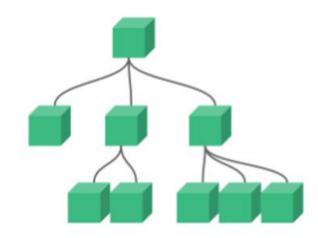
2. 一个单文件组件由三部分组成

template: 结构 (有且只能一个根元素)

script: js逻辑

style: 样式

- 3. 让组件支持 less
  - (1) style标签, lang= "less", 开启less功能
  - (2) 装包: yarn add less-loader@7.2.1 less -D





单文件组件有三个组成部分,哪三个?

template 结构

script js逻辑内容

style 样式





- ◆ vue基本概念
- ◆ @vue/cli 脚手架的使用
- ◆ vue插值表达式
- ◆ vue指令
- ◆ 综合案例 文章标题编辑



问题: vue是数据驱动的,应该如何提供数据,将来控制视图呢?



### vue如何提供数据

- 1. 通过 data 属性可以提供数据, data属性必须是一个函数
- 2. 这个函数需要返回一个对象,这个对象就代表vue提供的数据

```
<script>
export default {
    data () {
        return {
            money: 100,
            zs: 'zs'
        }
    }
}
</script>
```

使用插值表达式,可以在模板中渲染数据:



vue如何提供数据呢?

在 data 中提供, data是一个函数

函数的返回值是一个对象,对象就是vue提供的数据



### 插值表达式: 小胡子语法 {{ }}

- 1. 作用: 使用 data 中的数据渲染视图(模板)
- 2. 支持基本语法, 支持三元运算符

```
{{ msg }}
{{ obj.name }}
{{ msg.toUpperCase() }}
{{ obj.age > 18 ? '成年' : '未成年' }}
```

### 3. 注意点:

- (1) 使用数据在 data 中必须存在
- (2) 能使用表达式,但是不能使用语句 if for ...
- (3) 不能在标签属性中使用 {{ }} 插值



### 插值表达式的作用是什么?

使用 data 中的数据渲染视图(模板)

### 插值表达式的注意点是什么?

- (1) 使用数据在 data 中必须存在
- (2) 能使用表达式,但是不能使用语句 if for ...
- (3) 不能在标签属性中使用 {{ }} 插值



### 安装vue开发者工具:我们要调试vue应用,需要装插件。

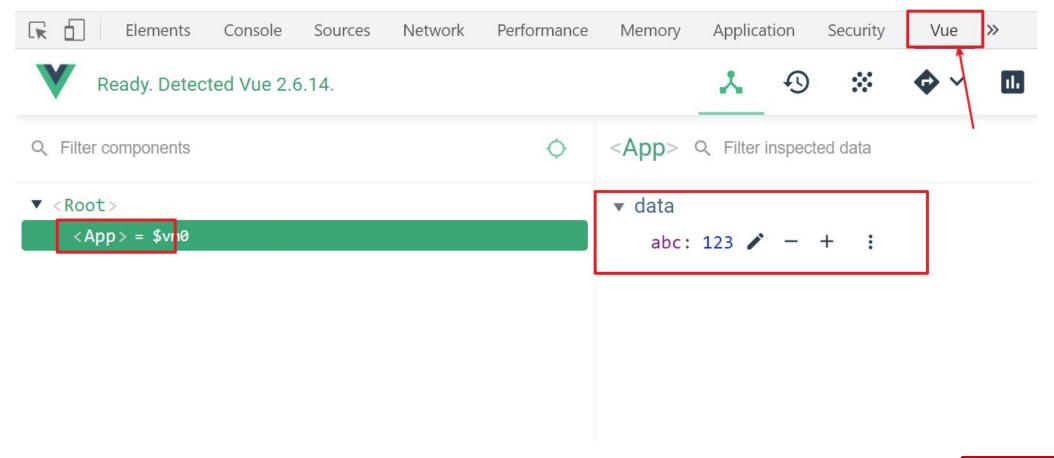
- 1. 直接通过谷歌应用商店安装 需要梯子
- 2. 通过极简插件下载插件,本地拖拽到chrome 扩展程序列表中,安装即可。 https://chrome.zzzmh.cn/index





### 安装vue开发者工具:

打开vue运行的页面,调试工具中Vue栏,即可查看修改数据,进行调试。







- ◆ vue基本概念
- ◆ @vue/cli 脚手架的使用
- ◆ vue插值表达式
- ◆ vue指令
- ◆ 综合案例 文章标题编辑



## vue指令: 特殊的 html 标签属性, 特点: v- 开头

每个 v- 开头的指令, 都有着自己独立的功能, 将来vue解析时, 会根据不同的指令提供不同的功能

# vue指令

```
<script>
export default {
    data() {
        return {
            words: '文字内容'
        }
    }
} </script>
```

文字内容





什么是 vue 指令?

vue指令:特殊的 html 标签属性,特点: v-开头



### vue指令-v-bind

说明:插值表达式不能用在html的属性上,想要动态的设置html元素属性,需要使用v-bind指令

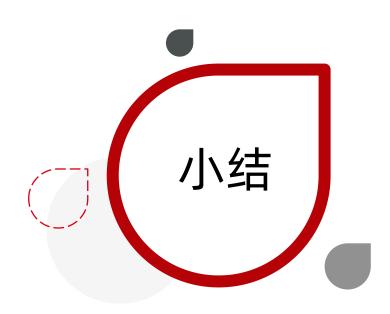
作用: 动态的设置html的标签属性

语法: v-bind:属性名="值"

简写: :属性名="值"

```
<!-- 完整语法 -->
<a v-bind:href="url"></a>
<!-- 缩写 -->
<a :href="url"></a>
```





1. v-bind指令的作用是什么?

动态的设置html的属性

2. v-bind语法是什么? 简写语法怎么写?

v-bind:属性名 = '值'

:属性名='值'(简写)





问题: 在vue中如何给按钮绑定点击事件呢?



### vue指令-v-on

作用: 注册事件

语法:

1. v-on:事件名="要执行的少量代码"

2. v-on:事件名= "methods中的函数名"

3. v-on:事件名= "methods中的函数名(实参)"

注意:事件处理函数在methods中提供

```
<hi>div
<h3>我是指令 - {{ money }}</h3>
<button v-on:click="money = money + 10">加十</button>
<button v-on:click="addMoney">加一</button>
<button v-on:click="sellHouse(5)">加五</button>
</div>
```

```
methods: {
   addMoney () {
      // methods中的所有函数都可以通过this访问到当前组件
      this.money++
   },
   sellHouse (price) {
      this.money += price
   }
}
```



### vue指令-v-on

作用: 注册事件

语法:

1. v-on:事件名="要执行的少量代码"

2. v-on:事件名= "methods中的函数名"

3. v-on:事件名= "methods中的函数名(实参)"

注意:事件处理函数在methods中提供

简写: v-on 可以 简写 成 @

```
<button @click="money = money + 10">加十</button>
  <button @click="addMoney">加一</button>
  <button @click="sellHouse(5)">加五</button>
```

```
methods: {
   addMoney () {
      // methods中的所有函数都可以通过this访问到当前组件
      this.money++
   },
   sellHouse (price) {
      this.money += price
   }
}
```





1. v-on指令的作用是什么?

### 注册事件

- 2. v-on指令的三个语法是什么?
  - (1) v-on:事件名= "要执行的少量代码"
  - (2) v-on:事件名= "methods中的函数名"
  - (3) v-on:事件名= "methods中的函数名(实参)"
- 3. v-on指令的简写语法是什么?
  - @事件名 = 'xxx'



默认a标签点击会跳走,希望阻止默认的跳转,

应该如何操作呢?如何阻止默认行为呢?

语法: e.preventDefault()

vue中如何获取事件对象呢?



### vue中获取事件对象(了解)

vue中获取事件对象:

- (1) 没有传参, 通过形参接收 e
- (2) 传参了, 通过\$event指代事件对象 e

```
<template>
 <div id="app">
   <a @click="fn" href="http://www.baidu.com">去百度</a>
    <a @click="fn2(100, $event)" href="http://www.baidu.com">去百度2</a>
 </div>
</template>
<script>
export default {
 methods: {
   fn(e) {
     e.preventDefault()
   },
   fn2(num, e) {
     e.preventDefault()
</script>
```





vue中如何获取事件对象呢?

- (1) 没有传参, 通过形参接收 e
- (2) 传参了, 通过\$event指代事件对象 e





e.preventDefault()单词很长不好写吧?

有没有一种更简单的方式实现呢?



### 事件修饰符

事件修饰符: vue提供事件修饰符,可以快速阻止默认行为或阻止冒泡

.prevent 阻止默认行为, .stop 阻止冒泡

```
<div id="app">
     <a @click.prevent="fn" href="http://www.baidu.com">去百度</a>
</div>
```





我想判断用户是否按下回车了怎么做?



### 按键修饰符:

在监听键盘事件时,我们经常需要判断详细的按键。可用按键修饰符。

需求: 用户输入内容, 回车时, 打印输入的内容。

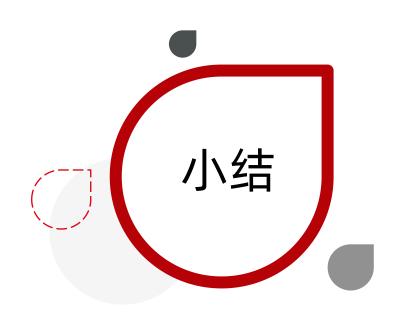
- @keyup.enter 监听回车键
- @keyup.esc 监听返回键

```
<div id="app">
     <input type="text" @keyup="fn"> <hr>
      <input type="text" @keyup.enter="fn2">
      </div>
```

### vue内置的按键修饰符列表:

- enter
- · .tab
- .delete (捕获"删除"和"退格"键)
- esc
- space
- up
- down
- left
- right





监听用户回车,是哪个修饰符?

@keyup.enter - 只要按下回车才能触发这个键盘事件函数





## 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)

## Hello, World

逆转世界

2





在 vue 中如何控制盒子的显示隐藏呢?



### v-show 和 v-if 功能: 控制盒子的显示隐藏

1. v-show

语法: v-show="布尔值" (true显示, false隐藏)

原理:实质是在控制元素的 css 样式, `display: none;`

2. v-if

语法: v-if="布尔值" (true显示, false隐藏)

原理: 实质是在动态的创建 或者 删除元素节点

#### 应用场景:

1. 如果是频繁的切换显示隐藏, 用 v-show

(v-show, 只是控制css样式, 而v-if, 频繁切换会大量的创建和删除元素, 消耗性能)

2. 如果是不用频繁切换, 要么显示, 要么隐藏的情况, 适合于用 v-if

(v-if 是惰性的, 如果初始值为 false, 那么这些元素就直接不创建了, 节省一些初始渲染开销)



vue如何控制标签元素显示/隐藏

v-show或v-if, 给变量赋值true/false, 显示/隐藏

区别是什么?

v-show是在控制元素的 css 样式

v-if 是动态的创建 或者 删除元素节点



### vue指令 - v-else 和 v-else-if

```
<div id="app">
 <h1 v-if="isLogin">尊敬的超级vip, 你好</h1>
 <h1 v-else>你谁呀,赶紧登陆~</h1>
 <hr>>
 <h1 v-if="age >= 60">60岁以上,广场舞</h1>
 <h1 v-else-if="age >= 30">30岁以上,搓麻将</h1>
 <h1 v-else-if="age >= 20">20岁以上,蹦迪</h1>
 <h1 v-else>20岁以下,唱跳rap篮球</h1>
</div>
```





## 折叠面板案例

一些实际网页效果:







百度旗下产品

展开٧

互联网巨头财报











facebook则 报

搜狐财报

阿里巴巴财 报

腾讯财报







谷歌财报

联想财报

陌陌财报



# **国** 案例

## 折叠面板

需求:点击展开或收起时,把面板内容区域显示或者隐藏

效果图如下:





## 1 案例

### 折叠面板

#### 步骤:

① 利用 v-show 指令, 控制底部面板的显示隐藏

```
<!-- 切换显示隐藏的部分 -->
<div v-show="isShow" class="container">
```

② 利用 v-on 指令, 注册点击事件, 点击时修改绑定的布尔值

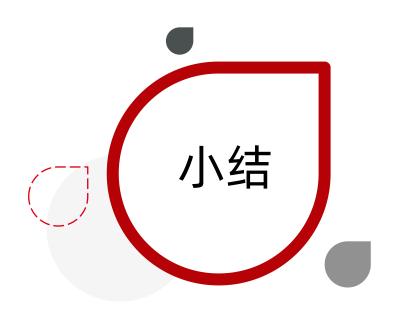
```
<!-- 按钮位置 -->
<span class="btn" @click="isShow = !isShow">
```

③ 根据 isShow, 利用三元运算符, 控制展开, 收起的文字切换

```
<!-- 按钮位置 -->
<span class="btn" @click="isShow = !isShow">
{{ isShow ? '收起' : '展开' }}
</span>
```







1. 控制 盒子显示隐藏 能想到哪个指令?

v-show

2. 给盒子 注册事件, 能想到哪个指令? 简写是什么?

v-on指令, 简写: @事件名

3. 控制页面文字的显示切换, 我们可以怎么操作?

插值表达式 {{ }} 配合 三元运算符?:



### v-model作用:给表单元素使用,双向数据绑定

什么是双向数据绑定?

- 1. 数据变化了, 视图会跟着变
- 2. 视图变化了, 数据要跟着变

语法: v-model='值'

<input type="text" v-model="msg">

用户名:



v-model 的作用是什么?

给表单元素使用,双向数据绑定



### vue指令 - v-model - 处理其他表单元素(了解)

### 例如:

- ① select
- ② checkbox
- ③ textarea

• • •

注意: v-model 会忽略掉表单元素原本的value, checked等初始值



### v-model 修饰符: 让v-model拥有更强大的功能

语法: v-model.修饰符="Vue数据变量"

● .number 转数字,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>
```





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



### vue指令-v-text和v-html

作用: 更新元素的innerText/innerHTML

- 语法:
  - v-text="值"
  - v-html="值"
- 区别:
  - v-text 不解析标签
  - v-html 解析标签

```
<template>
<div>
 </div>
</template>
```



v-text和v-html的区别是什么?

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





- ◆ vue基本概念
- ◆ @vue/cli 脚手架的使用
- ◆ vue插值表达式
- ◆ vue指令
- ◆ 综合案例 文章标题编辑



### 案例效果:

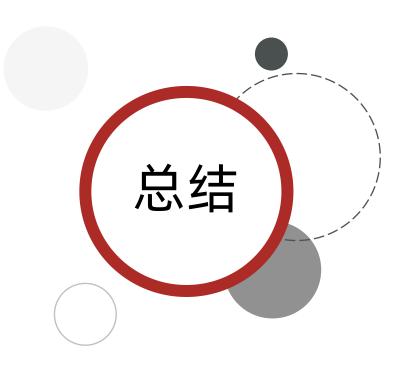
标题 测试 编辑

标题: 如何成为一名前端老鸟?

频道: 前端

取消

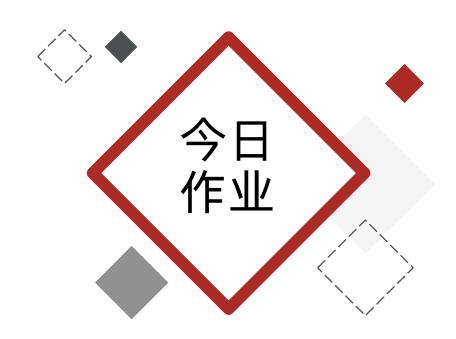
确认



了解vue是什么

知道vue-cli工具使用,能够创建一个vue项目

掌握vue的基础指令,能够完成页面基本交互



把课上例子,练习,案例做一遍

预习第二天内容



传智教育旗下高端IT教育品牌