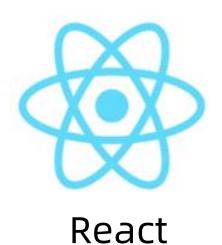
Vue 核心技术与实战











【Vue】前端必备岗位技能



多一句没有,少一句不行,用更短时间,教会更实用的技术!



岗位描述:

- 1、与设计师、产品紧密沟诵、深入理解产品需求
- 2、能理解产品的思路, 自我驱动完成产品模块
- 3、有效工作方式,提升团队的开发效率和产品质量;
- 4、负责前端技术框架的优化,以及前端性能优化,提供用户极速的响 岗位要求:
- 1、热爱前端, 热爱设计, 对新鲜事物充满好奇心, 有折腾的想法和精
- 2、熟悉前端常见的 类库及前端MVC,MVVM框架, 使用过 Vue React
- 3、思路清晰,逻辑思维强,优秀的沟通能力、责任心、执行力;
- 4、自我管理能力强良好、健康的创业心态、快速的学习能力、乐于分:
- 5、具备原生JS的开发能力,并且了解浏览器兼容性

加分项:

- 1、理解Vue, React, angularis等前端框架的核心设计思想
- 2、熟悉前后端分离开发模式,并且对Node服务器有一定的运维能力

前端开发经验

岗位职责:

- 1、负责游戏社区产品的前端开发;
- 2、负责游戏运营类中台系统的前端开发;
- 3、负责游戏各类网页活动的前端开发:

岗位要求:

- 1、学信网可查, 计算机或相关专业专科以上学历,
- 2. 熟悉ES6/CSS3/HTML5等开发调试技术:
- 3、熟练使用Vue + ElementUI开发中台系统;
- 4、熟悉组件化开发,能就活动中常用功能的组件
- 5、对产品交互和UI细节有严苛的要求
- 6、讨英语4级

职位描述

技术博客/开源项日有所贡献 Node.js React Vue 前端开发经验

岗位职责:

- 1、参与项目需求在用户端的实现讨论与方案确认;
- 2、完成可视化、前端应用、app/小程序等开发;

任职要求:

- 1、本科及以上学历:
- 2、基本功JS、CSS、HTML扎实;
- 3. 熟练使用Vue, 熟练使用ElementUI, AntDesign, iview等;
- 4、熟练使用Uniapp,了解RN、fullter等;
- 5、能提供参与项目的线上链接或本地演示

想学Vue,怎么学呢?



Vue2



Vue3

稳定

- 1. 最新长期稳定版 2.7.14
 2016年9月 六年多 近140次版本更新
- 2. 生态系统, 文档系统完备
- 3. 过往案例项目支撑

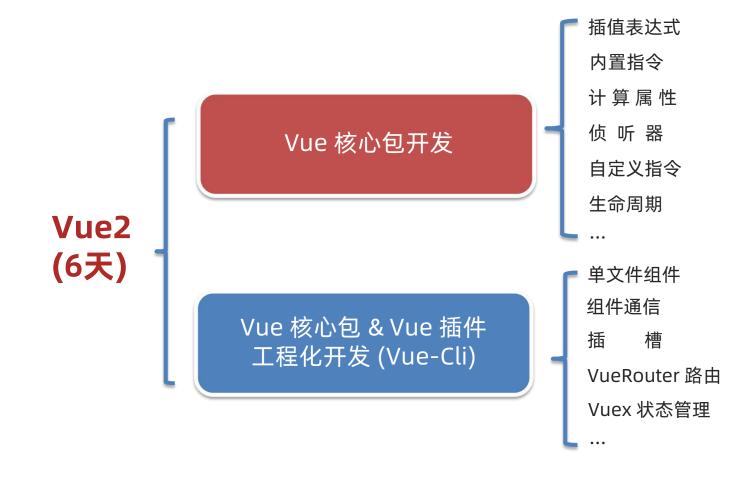
...

趋势

- 1 性能提升 => 1.2~2倍
- 2 内核体积更小 => 10kb
- 3 TS支持度更高
- 4组合式API, 封装复用逻辑

...









Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



综合案例:







Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



基于数据渲染出用户看到的页面

Vue 是什么

概念: Vue 是一个用于 构建用户界面^①的 渐进式^② 框架^③







凌美 (LAMY) 钢笔礼盒 宝... ¥488 00



爱他美 (Aptamil) 爱他美... ¥1878.00



¥209.00

title: '清仓大促', products: [{ id: 1, name: '手机', price: '1999元' }, { id: 2, name: '平板', price: '2999元' }, { id: 3, name: '电脑', price: '3999元' },]



清仓大促

- 手机 1999元
- 平板 2999元
- 电脑 3999元

用户页面(视图)



Vue 是什么

循序渐进

概念: Vue 是一个用于 构建用户界面^①的 渐进式^② 框架^③

Vuejs 核心包 声明式渲染 组件系统 客户端路由 大规模状态管理 构建工具 VueRouter Webpack/Vite Vuex

Vue 的两种使用方式:

① Vue 核心包开发

场景:局部模块改造

② Vue 核心包 & Vue 插件 工程化开发

场景:整站开发



一套完整的项目解决方案

Vue 是什么

概念: Vue 是一个用于 构建用户界面^①的 渐进式^② 框架^③

优点:大大提升开发效率 (70%↑)

缺点:需要理解记忆规则 → 官网



渐进式 JavaScript 框架





GITHUB

特别赞助



成为特别赞助商

易用

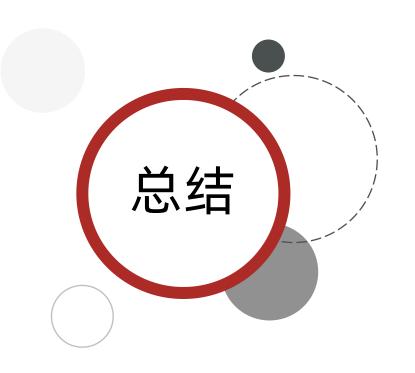
已经会了 HTML、CSS、 JavaScript? 即刻阅读指南开始构 建应用!

灵活

不断繁荣的生态系统,可以在一个库和一套完整框架之间自如伸缩。

高效

20kB min+gzip 运行大小 超快虚拟 DOM 最省心的优化



Vue是什么?

Vue 是一个用于 构建用户界面 的 渐进式 框架

1. 构建用户界面:基于数据动态渲染页面

2. 渐进式: 循序渐进的学习

3. 框架: 一套完整的项目解决方案,提升开发效率↑(理解记忆规则)

规则 → 官网





Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



创建 Vue 实例,初始化渲染

构建用户界面

创建 Vue 实例 初始化渲染 核心步骤 4步:

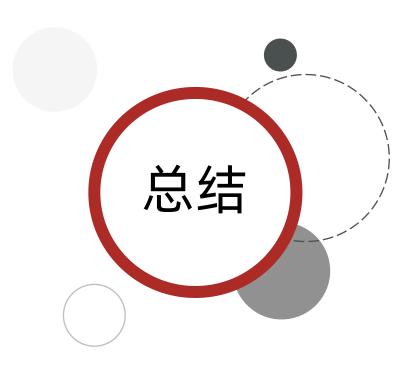
- 1. 准备容器
- 2. 引包 (官网) 开发版本 / 生产版本
- 3. 创建 Vue 实例 new Vue()
- 4. 指定配置项 → 渲染数据
 - ① el 指定挂载点
 - ② data 提供数据

Hello 黑马

msg: 'Hello 黑马'

```
<div id="app">
    {{ msg }}
</div>
<script src="https://cdn.jsdelivr.net/npm/vue@2.7.10/dist/vue.js"></script>

<script>
    const app = new Vue({
        el: '#app',
        data: {
              msg: 'Hello 黑马'
        }
    })
</script>
```



创建 Vue 实例,初始化渲染的核心步骤:

- 1. 准备容器
- 2. 引包(官网)-开发版本/生产版本
- 3. 创建 Vue 实例 new Vue()
- 4. 指定配置项 el data => 渲染数据
 - ① el 指定挂载点,选择器指定控制的是哪个盒子
 - ② data 提供数据





Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



插值表达式 {{ }}

插值表达式是一种 Vue 的模板语法

```
<div id="app">
{{ msg }}
</div>
```

```
data: {
   msg: 'Hello 黑马'
}
```

Hello 黑马



插值表达式 {{ }}

插值表达式是一种 Vue 的模板语法

1. 作用: 利用表达式进行插值, 渲染到页面中

表达式:是可以被求值的代码,JS引擎会将其计算出一个结果

```
money + 100
money - 100
money * 10
money / 10
```

```
price >= 100 ? '真贵' : '还行'
obj.name
arr[0]
fn()
obj.fn()
```



插值表达式 {{ }}

插值表达式是一种 Vue 的模板语法

1. 作用: 利用表达式进行插值, 渲染到页面中

表达式:是可以被求值的代码,JS引擎会将其计算出一个结果

2. 语法: {{ 表达式 }}

```
<h3>{{ title }}</h3>
{{ nickname.toUpperCase() }}
{{ age >= 18 ? '成年' : '未成年' }}
{{ obj.name }}
```

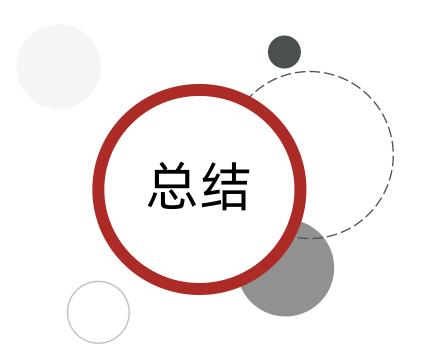
3. 注意点:

(1) 使用的数据必须存在 (data)

(2) 支持的是表达式,而非语句,比如: if for ...

(3) 不能在标签属性中使用 {{ }} 插值

```
我是p标签<//>
```



1. 插值表达式的作用是什么?

利用表达式进行插值,将数据渲染页面中

2. 语法格式?

{{ 表达式 }}

- 3. 插值表达式的注意点:
 - ① 使用的数据要存在 (data)
 - ② 支持的是表达式, 而非语句 if ... for
 - ③ 不能在标签属性里面使用





Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



Vue 核心特性:响应式

我们已经掌握了基础的模板渲染,其实除了基本的模板渲染,Vue背后还做了大量工作。

比如:数据的响应式处理 → 响应式:数据变化,视图自动更新

```
<div id="app">
  {{ msg }}
  </div>
```

```
const app = new Vue({
    el: '#app',
    data: { Hello Vue
        msg: 'Hello 黑马'
    } 响应式数据
})
```

Hello Vue Hello 黑马

如何访问 or 修改? data中的数据, 最终会被添加到实例上

① 访问数据: "实例.属性名"

② 修改数据: "实例.属性名" = "值"



Vue 核心特性:响应式

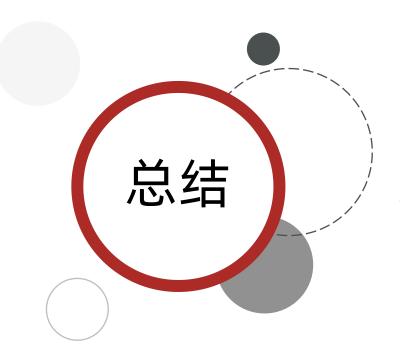
数据改变,视图会自动更新



聚焦于数据 → 数据驱动视图

使用 Vue 开发,关注业务的核心逻辑,根据业务修改数据即可





1. 什么是响应式呢?

数据改变,视图自动更新

使用 Vue 开发 → 专注于业务核心逻辑 即可

2. 如何访问或修改数据呢?

data中的数据, 最终会被添加到实例上

① 访问数据: "实例.属性名"

② 修改数据: "实例.属性名" = "值"





Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



安装 Vue 开发者工具:装插件调试 Vue 应用

- (1) 通过谷歌应用商店安装 (国外网站)
- (2) 极简插件: 下载 → 开发者模式 → 拖拽安装 → 插件详情允许访问文件

https://chrome.zzzmh.cn/index

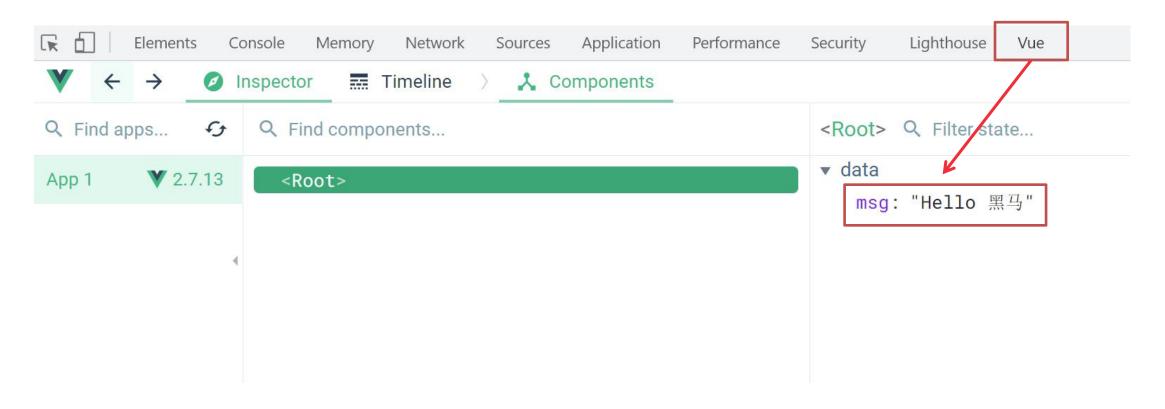






安装 Vue 开发者工具:装插件调试 Vue 应用

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 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本



Vue 指令

Vue 会根据不同的【指令】,针对标签实现不同的【功能】

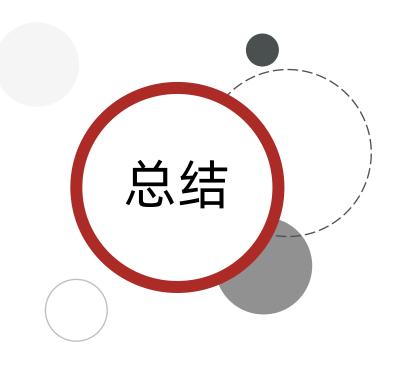
指令: 带有 v- 前缀 的 特殊 标签属性

```
<!-- Vue 指令: v- 前缀的标签属性 -->
<div v-html="str"></div>
```

```
<!-- 普通标签属性 -->
<div class="box"> ... </div>
<div title="小张"> ... </div>
```

作用:设置元素的 innerHTML

语法: v-html = "表达式 "



1. 什么是 Vue 指令呢?

指令就是带有 v- 前缀 的特殊 属性,不同属性 对应 不同的功能

2. 如果需要动态解析标签,可以用哪个指令?语法?

学习不同指令 → 解决不同业务场景需求

v-html = "表达式" → 动态设置元素 innerHTML



Vue 指令 v-show vs v-if

v-show

1. 作用: 控制元素显示隐藏

2. 语法: v-show = "表达式" 表达式值 true 显示, false 隐藏

3. 原理: 切换 display:none 控制显示隐藏

4. 场景: 频繁切换显示隐藏的场景



v-if

1. 作用: 控制元素显示隐藏(条件渲染)

2. 语法: v-if = "表达式" 表达式值 true 显示, false 隐藏

3. 原理: 基于条件判断,是否 创建 或 移除 元素节点

4. 场景: 要么显示,要么隐藏,不频繁切换的场景





Vue 指令 v-else v-else-if

1. 作用: 辅助 v-if 进行判断渲染

2. 语法: v-else v-else-if = "表达式"

3. 注意: 需要紧挨着 v-if 一起使用

♂男

♀女

成绩评定A: 奖励电脑一台

成绩评定B: 奖励周末郊游

成绩评定C: 奖励零食礼包

成绩评定D: 惩罚一周不能玩手机

性别	性别标识
男	1
女	2

序号	成绩等级	考试分数
1	Α	90分+
2	В	70-90分
3	С	60-70分
4	D	60分以下



Vue 指令 v-on

1.作用: 注册事件 = 添加监听 + 提供处理逻辑

2. 语法:

① v-on:事件名 = "内联语句"

② v-on:事件名 = "methods中的函数名"

3. 简写: @事件名

<button v-on:click="count++">按钮</button>

<button @click="count++">按钮</button>



Vue 指令 v-on

1.作用: 注册事件 = 添加监听 + 提供处理逻辑

2. 语法:

① v-on:事件名 = "内联语句"

② v-on:事件名 = "methods中的函数名"

3. 简写: @事件名

```
<button @click="fn">-</button>
```

4. 注意: methods函数内的 this 指向 Vue 实例

```
const app = new Vue({
 el: '#app',
 data: {
   // 提供数据
   count: 100
 methods: {
   // 提供处理逻辑函数
   fn () {
     console.log('提供逻辑代码')
```



Vue 指令 v-on 调用传参

```
<button @click="fn">
按钮
</button>
```

```
const app = new Vue({
   el: '#app',
   methods: {
     fn () {
        console.log('这是一个fn函数')
      }
   }
}
```



Vue 指令 v-on 调用传参

```
<button @click="fn()">
按钮
</button>
```

```
const app = new Vue({
   el: '#app',
   methods: {
     fn () {
        console.log('这是一个fn函数')
      }
   }
}
```



Vue 指令 v-on 调用传参

```
<button @click="fn(参数1,参数2)">
按钮
</button>
```

```
const app = new Vue({
   el: '#app',
   methods: {
     fn () {
        console.log('这是一个fn函数')
      }
   }
})
```



Vue 指令 v-on 调用传参

```
<button @click="fn(参数1,参数2)">
按钮
</button>
```

小黑自动售货机

可乐5元

咖啡10元

银行卡余额: 100元

```
const app = new Vue({
   el: '#app',
   methods: {
     fn (a, b) {
        console.log('这是一个fn函数')
     }
   }
})
```



Vue 指令 v-bind

1.作用: 动态的设置html的标签属性 → src url title ...

2. 语法: v-bind:属性名="表达式"

3. 注意: 简写形式:属性名="表达式"

```
const app = new Vue({
   el: '#app',
   data: {
      url: '图片路径'
   }
})
```





图片切换案例-波仔学习之旅





图片切换案例-波仔学习之旅



核心思路分析:

- ① 数组存储图片路径 → [图片1,图片2,图片3,...]
- ② 准备下标 index,数组[下标] \rightarrow v-bind 设置 src 展示图片 \rightarrow 修改下标切换图片



Vue 指令 v-for

1.作用: 基于数据循环, 多次渲染整个元素 → 数组、对象、数字...

2. 遍历数组语法:

v-for = "(item, index) in 数组"

- ▶ item 每一项, index 下标
- ➤ 省略 index: v-for = "item in 数组"



图书管理案例 - 小黑的书架

明确需求:

- ① 基本渲染 → v-for
- ② 删除功能 → 用 filter 根据 id 从数组中删除对应项

小黑的书架

- 《红楼梦》 曹雪芹
- 《西游记》 吴承恩 删除
- 《水浒传》 施耐庵 删除
- 《三国演义》 罗贯中 删除

删除



语法: key属性 = "唯一标识"

作用:给列表项添加的唯一标识。便于Vue进行列表项的正确排序复用。

```
     <!i v-for="(item, index) in booksList" :key="item.id">
          <span>{{ item.name }}</span>
          <span>{{ item.author }}</span>
          <button @click="del(item.id)">删除</button>
```

小黑的书架

- 《红楼梦》 曹雪芹 删除
- 《西游记》 吴承恩 删除
- 《水浒传》 施耐庵 删除
- 《三国演义》 罗贯中 删除



key作用:给元素添加的唯一标识。

```
booksList: [
    { id: 1, name: '《红楼梦》', author: '曹雪芹' },
    { id: 2, name: '《西游记》', author: '吴承恩' },
    { id: 3, name: '《水浒传》', author: '施耐庵' },
    { id: 4, name: '《三国演义》', author: '罗贯中' }
]
```

```
:key="item.id"
```



$$key = "1"$$

$$key = "2"$$

$$key = "3"$$



key作用:给元素添加的唯一标识。

```
booksList: [
    { id: 2, name: '《西游记》', author: '吴承恩' },
    { id: 3, name: '《水浒传》', author: '施耐庵' },
    { id: 4, name: '《三国演义》', author: '罗贯中' }
]
```

```
:key="item.id"
```

红楼梦 曹雪芹

key = "1"

西游记 吴承恩

key = "2"

水浒传 施耐庵

三国演义 罗贯中



key作用:给元素添加的唯一标识。

```
booksList: [
    { id: 2, name: '《西游记》', author: '吴承恩' },
    { id: 3, name: '《水浒传》', author: '施耐庵' },
    { id: 4, name: '《三国演义》', author: '罗贯中' }
]
```

```
:key="item.id"
```



v-for 中的 key - 不加 key

v-for 的默认行为会尝试 原地修改元素 (就地复用)

```
booksList: [
    { id: 2, name: '《西游记》', author: '吴承恩' },
    { id: 3, name: '《水浒传》', author: '施耐庵' },
    { id: 4, name: '《三国演义》', author: '罗贯中' }
]
```



v-for 中的 key - 不加 key

v-for 的默认行为会尝试 原地修改元素 (就地复用)

```
booksList: [
    { id: 2, name: '《西游记》', author: '吴承恩' },
    { id: 3, name: '《水浒传》', author: '施耐庵' },
    { id: 4, name: '《三国演义》', author: '罗贯中' }
]
```

西游记 吴承恩

水浒传 施耐庵

三国演义 罗贯中



key作用:

给元素添加的唯一标识,便于Vue进行列表项的正确排序复用。

注意点:

- 1. key 的值只能是 字符串 或 数字类型
- 2. key 的值必须具有 唯一性
- 3. 推荐使用 id 作为 key(唯一),不推荐使用 index 作为 key(会变化,不对应)

v-for="(item, index) in xxx" :key="唯一值">



Vue 指令 v-model

- 1. 作用: 给 表单元素 使用, 双向数据绑定 → 可以快速 获取 或 设置 表单元素内容
 - ① 数据变化 → 视图自动更新
 - ② 视图变化 → 数据自动更新
- **2. 语法:** v-model = '变量'







◆ Vue 快速上手

Vue 概念 / 创建实例 / 插值表达式 / 响应式特性 / 开发者工具

◆ Vue 指令

v-html / v-show / v-if / v-else / v-on / v-bind / v-for / v-model

◆ 综合案例 - 小黑记事本

列表渲染/删除功能/添加功能/底部统计/清空



综合案例 - 小黑记事本

功能需求:

- ①列表渲染
- ② 删除功能
- ③ 添加功能
- ④ 底部统计 和 清空







功能总结:

① 列表渲染:

v-for key 的设置 {{ }} 插值表达式

②删除功能

v-on 调用传参 filter 过滤 覆盖修改原数组

③ 添加功能

v-model 绑定 unshift 修改原数组添加

④ 底部统计 和 清空

数组.length累计长度

覆盖数组清空列表

v-show 控制隐藏



传智教育旗下高端IT教育品牌