

邂逅Vue3开发

王红元 coderwhy

自我介绍



王红元
coderwhy

《TypeScript数据结构与算法》《React技术栈权威指南》作者

腾讯AI高校训练营多所985、211名牌大学特聘讲师。

丰富的软件开发和教学经验，带领团队开发出众多公司大型项目和软件系统。

全面负责国内知名视频直播软件<新浪秀场>项目架构和各端的设计和开发。

澳大利亚《智慧养老系统》研发负责人、CTO。

带领团队开发欧洲社交软件得心App。

课前提醒

■ 上课时间：

- 一般情况下，周一三五的晚上 20:00 ~ 22:00；
- 如果因为放假或者特殊情况修改了时间，会在群里通知大家；
- 建议来听直播，直播会更有感觉；

■ 上课的环境：

- 上课我用的是Mac电脑，但是所有的操作和Windows电脑上是一致的（有不一致的我会专门录制视频）；
- 上课我会使用VSCode作为开发工具，你也可以选择webstorm等其他工具；

■ 相互尊重、共同进步：

- 每个同学的基础不一样，之前的学习经历和方向不同；
- 所讲的内容是为大部分同学考虑的，希望大家可以相互理解、相互帮助、共同进步；

■ 课程资料：每次上完课，会将资料上传到腾讯课堂。



- Vue (读音 /vju:/ , 类似于 **view**) 是一套用于构建用户界面的**渐进式框架**。
 - 全程是Vue.js或者Vuejs ;
 - 什么是渐进式框架呢？表示我们可以在项目中一点点来引入和使用Vue，而不一定需要全部使用Vue来开发整个项目；

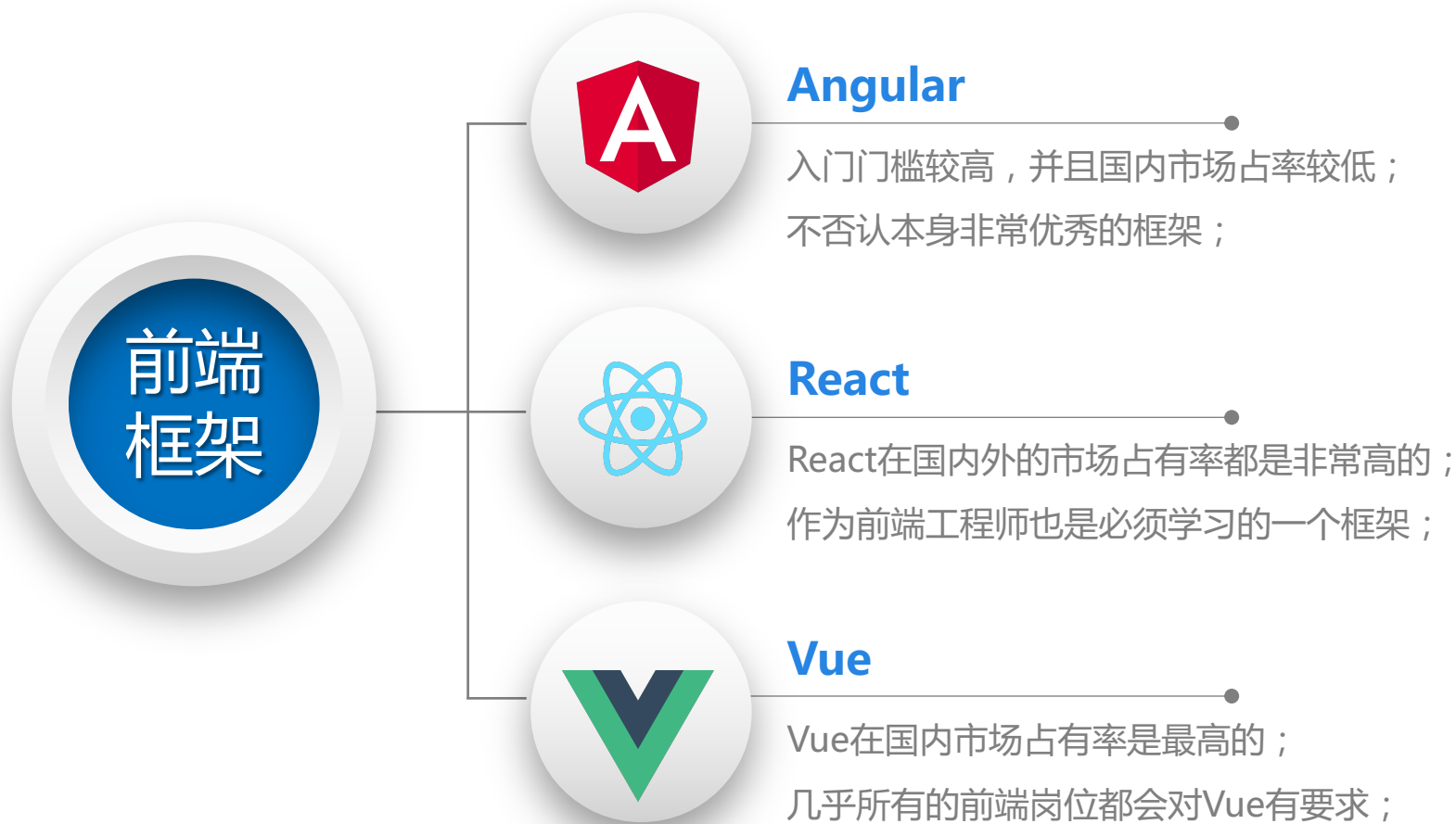


渐进式 JavaScript 框架

[WHY VUE.JS?](#)[起步](#)[GITHUB](#)

目前Vue在前端处于什么地位？

- 目前前端最流行的是三大框架：Vue、React、Angular。



框架数据对比（Google指数）

vue
搜索字词

react
搜索字词

angular
搜索字词

+ 添加比较项

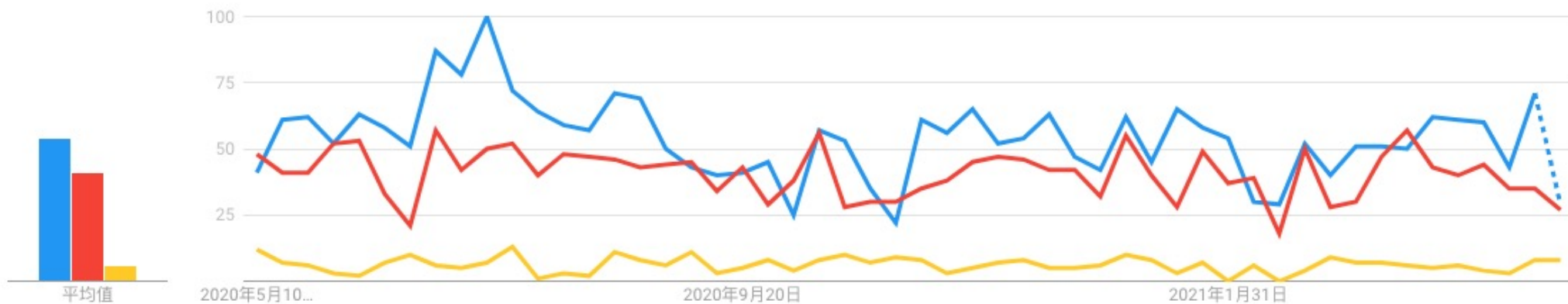
中国 ▼

过去 12 个月 ▼

所有类别 ▼

Google 网页搜索 ▼

热度随时间变化的趋势 ?



框架数据对比（百度指数）

关键词

vue

react

angular

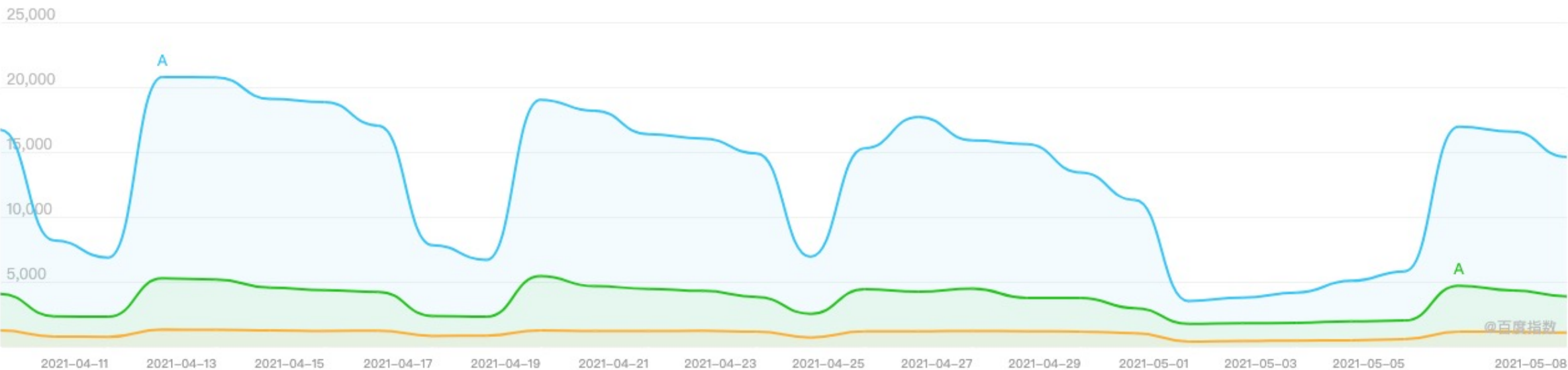
+ 添加对比

确定

搜索指数 ?

☐ 对比时间段 | 2021-04-09 ~ 2021-05-08 | 近30天 | PC+移动 | 全国 |

vue react angular

☒ 新闻头条 ☐ 平均值

2011

2012

2013

2014

2015

2016

2017

2018

2019

2020

2021

框架数据对比（npm下载量）

vue vs react vs angular

Enter an npm package...

vue x

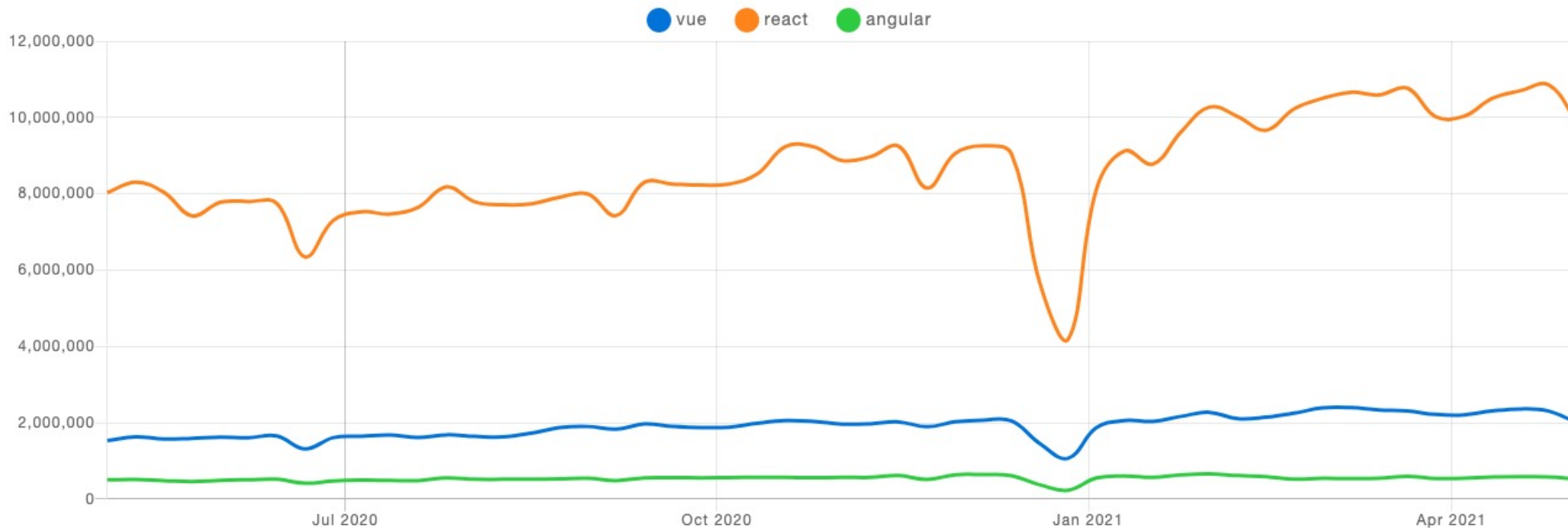
react x

angular x

+ @angular/core

+ ember-source

Downloads in past 1 Year ▾



框架数据对比 (GitHub)

vuejs/vue

👉 Vue.js is a progressive, incrementally-adoptable JavaScript framework for building UI on the web.

vue javascript framework frontend

☆ 183k ● JavaScript MIT license Updated 16 hours ago 1 issue needs help

facebook/react

A declarative, efficient, and flexible JavaScript library for building user interfaces.

react javascript library ui frontend declarative

☆ 168k ● JavaScript MIT license Updated 12 hours ago 9 issues need help

angular/angular

One framework. Mobile & desktop.

angular javascript typescript web-performance web pwa web-framework

☆ 73k ● TypeScript MIT license Updated 7 hours ago 35 issues need help



谁是最好的前端框架？

■ 当然，我不会去给出我的结论：

- 首先，这是一个敏感的话题，在很多地方都争论不休，就像很多人喜欢争论谁才是世界上最好的语言一样；
- 其次，争论这个话题是没有意义的，争论不休的话题；

■ 但是，我们从现实的角度，分析一下，学习哪一门语言更容易找到工作？

- 找后端的工作：优先推荐Java、其次推荐Go、再次推荐Node (JavaScript)，可能不推荐PHP、C#；
- 找前端的工作：优先推荐JavaScript (TypeScript)、其次Flutter、再次Android (Java、Kotlin)、iOS (OC、Swift)；
- 也有很多的其他方向：游戏开发、人工智能、算法工程师等等；

■ 那么，就前端来说，学习了HTML、CSS、JavaScript，哪一个框架更容易找到工作？

- 如果去国外找工作，优先推荐React、其次是Vue和Angular，不推荐jQuery了；
- 如果在国内找工作，优先推荐、必须学习Vue，其次是React，其次是Angular，不推荐jQuery了；

Vue对前端工程师的重要性

学好Vue你一定可以找到一份满意的前端工作

如果你没有掌握Vue很难找到一份满意的前端工作



前端岗位

学习vue2还是vue3 ?

《程序员》：Vue 3 版本兼容 2.x，对于想要学习 Vue 的开发者而言，时常在纠结是从 Vue 2 开始学基础还是直接学 Vue 3，对此，你有着什么样的建议？

尤雨溪：直接学 Vue 3 就行了，基础概念是一模一样的。

目前需要学习Vue3吗？

- 在2020年的9月19日，万众期待的Vue3终于发布了正式版，命名为 **“One Piece”** 。
 - 它也带来了很多新的特性：**更好的性能、更小的包体积、更好的TypeScript集成、更优秀的API设计。**
 - 在vue3刚刚发布时，很多人也是跃跃欲试，想要尝试vue3的各种新特性。
 - 但是事实上在刚刚发布的时候我们使用vue3来写demo练习是没有问题的，真正在实际业务项目中使用vue3还需要一个相对的过程；
 - 包括vue3的进一步稳定、包括社区更多vue3相关的插件、组件库的支持和完善。
- 那么现在是否是学习vue3的时间呢？
 - **答案是肯定的**
 - 首先vue3在经过一系列的更新和维护后，已经是趋于稳定，并且在之前尤雨溪也宣布在**今年（2021年）第二季度**会将vue3作为Vue CLI的默认版本了。
 - 目前社区也经过一定时间的沉淀，更加的完善了，包括**AntDesignVue、Element-Plus**都提供了对Vue3的支持，所以很多公司目前新的项目都已经在使用Vue3来进行开发了。
 - 并且在面试的时候，几乎都会问到各种各样Vue3、Vite2工具相关的问题。

Vue3带来的变化（源码）

■ 源码通过monorepo的形式来管理源代码：

- Mono：单个

- Repo：repository仓库

- 主要是将许多项目的代码存储在同一个repository中；

- 这样做的目的是多个包本身相互独立，可以有自己的功能逻辑、单元测试等，同时又在同一个仓库下方便管理；

- 而且模块划分的更加清晰，可维护性、可扩展性更强；

■ 源码使用TypeScript来进行重写：

- 在Vue2.x的时候，Vue使用Flow来进行类型检测；

- 在Vue3.x的时候，Vue的源码全部使用TypeScript来进行重构，并且Vue本身对TypeScript支持也更好了；



Vue3带来的变化（性能）

■ 使用Proxy进行数据劫持

- 在Vue2.x的时候，Vue2是使用Object.defineProperty来劫持数据的getter和setter方法的；
- 这种方式一致存在一个缺陷就是当给对象添加或者删除属性时，是无法劫持和监听的；
- 所以在Vue2.x的时候，不得不提供一些特殊的API，比如\$set或\$delete，事实上都是一些hack方法，也增加了开发者学习新的API的成本；
- 而在Vue3.x开始，Vue使用Proxy来实现数据的劫持，这个API的用法和相关的原理我也会在后续讲到；

■ 删除了一些不必要的API：

- 移除了实例上的\$on, \$off 和 \$once；
- 移除了一些特性：如filter、内联模板等；

■ 包括编译方面的优化：

- 生成Block Tree、Slot编译优化、diff算法优化；



Vue3带来的变化（新的API）

■ 由Options API 到 Composition API：

- 在Vue2.x的时候，我们会通过Options API来描述组件对象；
- Options API包括data、props、methods、computed、生命周期等等这些选项；
- 存在比较大的问题是多个逻辑可能是在不同的地方：
 - ✓ 比如created中会使用某一个method来修改data的数据，代码的内聚性非常差；
- Composition API可以将 相关联的代码 放到同一处 进行处理，而不需要在多个Options之间寻找；

■ Hooks函数增加代码的复用性：

- 在Vue2.x的时候，我们通常通过mixins在多个组件之间共享逻辑；
- 但是有一个很大的缺陷就是 mixins也是由一大堆的Options组成的，并且多个mixins会存在命名冲突的问题；
- 在Vue3.x中，我们可以通过Hook函数，来将一部分独立的逻辑抽取出去，并且它们还可以做到是响应式的；
- 具体的好处，会在后续的课程中演练和讲解（包括原理）；

系统学习Vue3+TypeScript





如何使用Vue呢？

- Vue的本质，就是一个JavaScript的库：
 - 刚开始我们不需要把它想象的非常复杂；
 - 我们就把它理解成一个已经帮助我们封装好的库；
 - 在项目中可以引入并且使用它即可。

- 那么安装和使用Vue这个JavaScript库有哪些方式呢？
 - 方式一：在页面中通过CDN的方式来引入；
 - 方式二：下载Vue的JavaScript文件，并且自己手动引入；
 - 方式三：通过npm包管理工具安装使用它（webpack再讲）；
 - 方式四：直接通过Vue CLI创建项目，并且使用它；

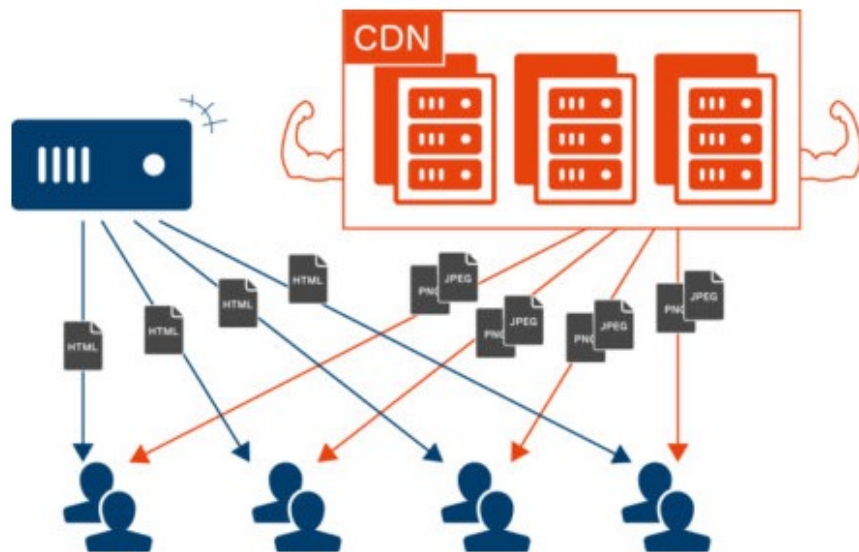
方式一：CDN引入

■ 什么是CDN呢？CDN称之为内容分发网络（**Content Delivery Network**或**Content Distribution Network**，缩写：**CDN**）

- 它是指通过 **相互连接的网络系统**，利用**最靠近**每个用户的服务器；
- 更快、更可靠地将**音乐、图片、视频、应用程序及其他文件**发送给用户；
- 来提供**高性能、可扩展性及低成本**的网络内容传递给用户；

■ 常用的CDN服务器可以大致分为两种：

- 自己的CDN服务器：需要购买自己的CDN服务器，目前阿里、腾讯、亚马逊、Google等都可以购买CDN服务器；
- 开源的CDN服务器：国际上使用比较多的是unpkg、JSDelivr、cdnjs；



方式一：CDN引入

■ Vue的CDN引入：

```
<script src="https://unpkg.com/vue@next"></script>
```

■ Hello Vue案例的实现：

```
<div id="app"></div>

<script src="https://unpkg.com/vue@next"></script>
<script>
  // Vue相关的代码
  const app = Vue.createApp({
    template: '<h2>Hello Vue3</h2>'
  });

  // 将app挂载到id为app的div上
  app.mount("#app");
</script>
```

方式二：下载和引入

■ 下载Vue的源码，可以直接打开CDN的链接：

□ 打开链接，复制其中所有的代码；

□ 创建一个新的文件，比如vue.js，将代码复制到其中；

■ 通过script标签，引入刚才的文件：

```
<script src="../../js/vue.js"></script>
```

■ 你好啊，Vue3，案例的实现：

```
<div id="app"></div>

<script src="../../js/vue.js"></script>
<script>
  const app = Vue.createApp({
    template: `<h2>你好啊，Vue3</h2>`
  });

  app.mount('#app');
</script>
```

计数器案例

- 如果我们希望实现一个计数器的案例：
 - 点击+1，那么内容会显示数字+1；
 - 点击-1，那么内容会显示数字-1；
- 我们可以选择很多种方式来实现：
 - 在这里我们就对比原生和Vue的实现方式的不同。

计数器原生实现

```
// 1. 获取DOM原生
const titleEl = document.querySelector('.title');
const btnInEl = document.querySelector('.increment');
const btnDeEl = document.querySelector('.decrement');

// 2. 默认设置的是Hello World
let counter = 0

// 3. 设置titleEl的内容
titleEl.innerHTML = counter;

// 4. 监听按钮的点击
btnInEl.addEventListener('click', () => {
  counter += 1;
  titleEl.innerHTML = counter;
})
btnDeEl.addEventListener('click', () => {
  counter -= 1;
  titleEl.innerHTML = counter;
})
```

计数器Vue实现

```
<div id="app"></div>
<template id="my-app">
  <div>
    <h2>{{counter}}</h2>
    <button @click='increment'>+1</button>
    <button @click='decrement'>-1</button>
  </div>
</template>
<script src="../../js/vue.js"></script>
```

```
<script>
  const App = {
    template: "#my-app",
    data() {
      return {
        counter: 0
      }
    },
    methods: {
      increment() {
        this.counter++;
      },
      decrement() {
        this.counter--;
      }
    }
  }

  Vue.createApp(App).mount('#app');
</script>
```


声明式和命令式

■ 原生开发和Vue开发的模式和特点，我们会发现是完全不同的，这里其实涉及到**两种不同的编程范式**：

□ **命令式编程**和**声明式编程**；

□ 命令式编程关注的是 “**how to do**”，声明式编程关注的是 “**what to do**”，由**框架(机器)完成** “**how**” 的过程；

■ **在原生的实现过程中，我们是如何操作的呢？**

□ 我们每完成一个操作，都需要通过**JavaScript编写一条代码**，来给**浏览器一个指令**；

□ 这样的编写代码的过程，我们称之为**命令式编程**；

□ 在早期的原生JavaScript和jQuery开发的过程中，我们都是通过这种命令式的方式在编写代码的；

■ **在Vue的实现过程中，我们是如何操作的呢？**

□ 我们会在createApp传入的对象中声明需要的内容，模板template、数据data、方法methods；

□ 这样的编写代码的过程，我们称之为是**声明式编程**；

□ 目前Vue、React、Angular的编程模式，我们称之为声明式编程；

MVVM模型

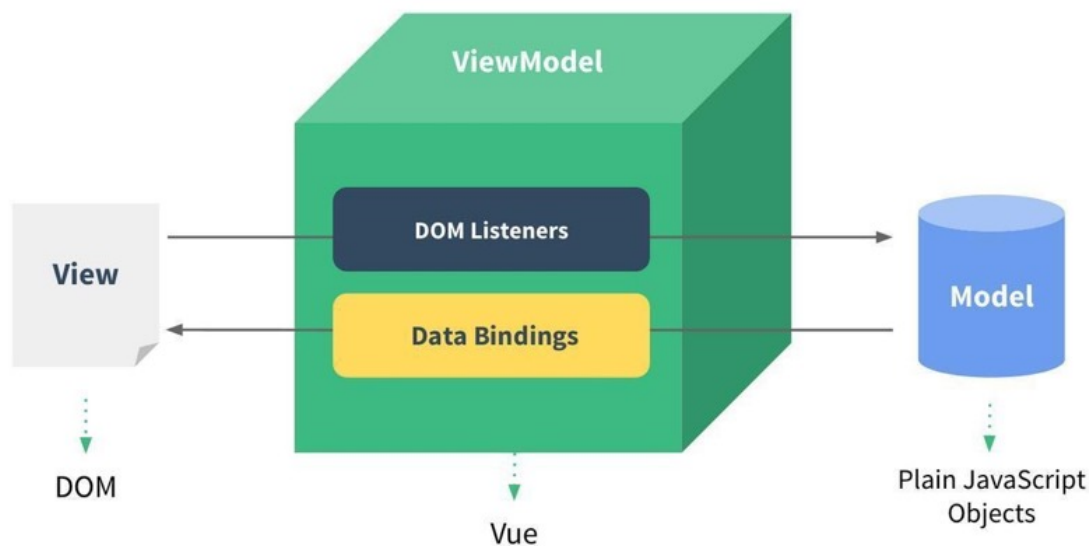
■ MVC和MVVM都是一种软件的体系结构

□ MVC是Model – View –Controller的简称，是在前期被使用非常框架的架构模式，比如iOS、前端；

□ MVVM是Model-View-ViewModel的简称，是目前非常流行的架构模式；

■ 通常情况下，我们也经常称Vue是一个MVVM的框架。

□ Vue官方其实有说明，Vue虽然并没有完全遵守MVVM的模型，但是整个设计是受到它的启发的。





template属性

- 在使用createApp的时候，我们传入了一个**对象**，接下来我们详细解析一下之前传入的属性分别代表什么含义。
- **template属性**：表示的是Vue需要帮助我们渲染的模板信息：
 - 目前我们看到它里面**有很多的HTML标签**，这些标签会**替换掉**我们挂载到的元素（比如id为app的div）的innerHTML；
 - 模板中有一些**奇怪的语法**，比如 {{}}，比如 @click，这些都是**模板特有的语法**，我们会在后面讲到；
- 但是这个模板的写法**有点过于别扭**了，并且**IDE很有可能没有任何提示**，阻碍我们编程的效率。
- **Vue提供了两种方式**：
 - 方式一：使用script标签，并且标记它的类型为 x-template；
 - 方式二：使用任意标签（通常使用template标签，因为不会被浏览器渲染），设置id；
 - ✓ template元素是一种用于保存客户端内容的机制，该内容再加载页面时不会被呈现，但随后可以在运行时使用JavaScript实例化；

■ 方式一：使用script标签

```
<script type="x-template" id="my-app">
  <div>
    <h2>{{counter}}</h2>
    <button @click='increment'+1</button>
    <button @click='decrement'-1</button>
  </div>
</script>
```

■ 方式二：使用template

```
<template id="my-app">
  <div>
    <h2>{{counter}}</h2>
    <button @click='increment'+1</button>
    <button @click='decrement'-1</button>
  </div>
</template>
```

■ 这个时候，在createApp的对象中，我们需要传入的template以 # 开头：

□ 如果字符串是以 # 开始，那么它将被用作 querySelector，并且使用匹配元素的 innerHTML 作为模板字符串；



data属性

- **data属性**是传入一个函数，并且该函数需要返回一个对象：
 - 在Vue2.x的时候，也可以传入一个对象（虽然官方推荐是一个函数）；
 - 在Vue3.x的时候，必须传入一个函数，否则就会直接在浏览器中报错；
- data中返回的对象会被**Vue的响应式系统劫持**，之后对该**对象的修改或者访问**都会在劫持中被处理：
 - 所以我们在template中通过 `{{counter}}` 访问counter，可以从对象中获取到数据；
 - 所以我们修改counter的值时，template中的 `{{counter}}`也会发生改变；
- 具体这种响应式的原理，我们后面会有专门的篇幅来讲解。

- **methods属性**是一个对象，通常我们会在这个对象中定义很多的方法：
 - 这些方法可以被绑定到 **template 模板**中；
 - 在该方法中，我们可以使用**this关键字**来直接访问到data中返回的对象的属性；
- 对于有经验的同学，在这里我提一个问题，官方文档有这么一段描述：
 - 问题一：为什么不能使用**箭头函数**（官方文档有给出解释）？
 - 问题二：不使用箭头函数的情况下，**this到底指向的是什么**？（可以作为一道面试题）



注意

注意，不应该使用箭头函数来定义 **method 函数** (例如 `plus: () => this.a++`)。理由是箭头函数绑定了父级作用域的上下文，所以 `this` 将不会按照期望指向组件实例，`this.a` 将是 `undefined`。

- 当然，这里还可以定义很多其他的属性，我们会在后续进行讲解：
 - 比如props、computed、watch、emits、setup等等；
 - 也包括很多的生命周期函数；
- 不用着急，我们会一个个学习它们的。



Vue的源码

■ 如果想要学习Vue的源码，比如看createApp的实现过程，应该怎么办呢？

■ 第一步：在GitHub上搜索 vue-next，下载源代码；

□ 这里推荐通过 git clone 的方式下载；

■ 第二步：安装Vue源码项目相关的依赖；

□ 执行 yarn install

■ 第三步：对项目执行打包操作

□ 执行 yarn dev（执行前修改脚本）

```
"scripts": {  
  "dev": "node scripts/dev.js --sourcemap",  
}
```

■ 第四步：通过 packages/vue/dist/vue.global.js 调试代码