



第三章 UNIAPP 与 VUE (一)



目录

- 一. Uniapp与Vue的学习路径
- 二. Vue概述
- 一. 三 Vue开发与传统前端开发的不同
- 三. 四、Uniapp页面

一、UNIAPP与VUE的学习路径

■从uniapp入手

- 学习资源：Uniapp》教程》vue》vue3语法（左图）
[介绍 | uni-app官网 \(dcloud.net.cn\)](#)

■从Vue官网入手

- <https://cn.vuejs.org/>
- 文档》深度指南（右图）
- 提供演练场

■从案例出发，课程讲授将从多角度综合展开

■需要具备的基础知识

- JS：具备JS基础知识，包括关键字、语法、事件对象引用等。
- 必要的H5与CSS用法



二、VUE概述

1. Vue是什么
2. 三大前端框架：VUE、React、Angular
3. Vue的安装方法
4. SFC

2.1 VUE

渐进式 JavaScript 框架

- 是一款用于构建用户界面的 JavaScript 框架。它基于标准 HTML、CSS 和 JavaScript 构建，并提供了一套**声明式的、组件化的**编程模型，帮助你高效地开发用户界面。
 - **声明式渲染**：Vue 基于标准 HTML 拓展了一套**模板语法**，使得我们可以**声明式地描述**最终输出的 HTML 和 JavaScript 状态之间的关系。
 - **响应性**：Vue 会自动跟踪 JavaScript 状态并在**其发生变化时响应式地更新** DOM。

2.1 VUE是一种MVVM

■ VUE是一种MVVM (Model-View-ViewModel: 模型-视图-视图模型)

- Model: 代表数据模型
- View: 只专注视图UI处理
- ViewModel: 只处理业务和数据。
- MVVM的核心是 “VM” 。 ViewModel 负责连接 View 和 Model, 保证视图和数据的一致性。
- 这种轻量级的架构让前端开发更加高效、便捷, 大幅减少代码行数, 同时增量渲染性能更好。

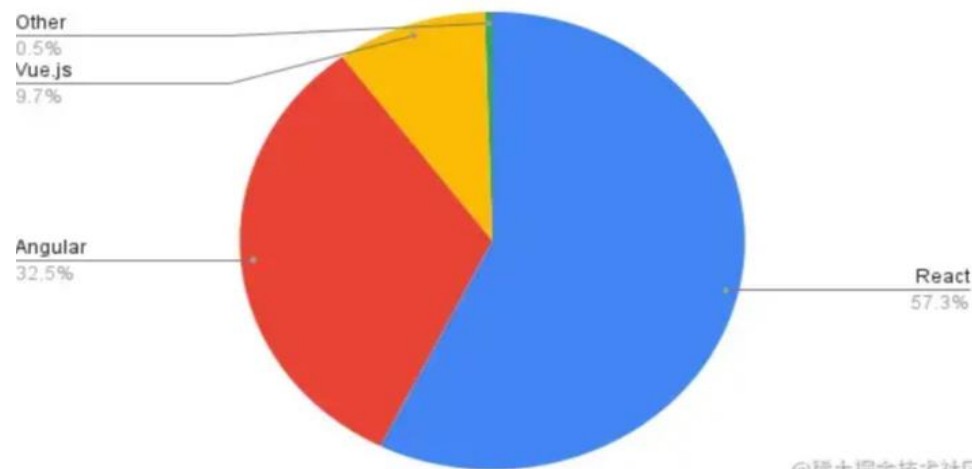
2.2 三大前端框架：VUE、REACT、ANGULAR

■ React: Facebook (Instagram)

■ Angular : Google

■ VUE: 尤雨溪 (Evan You): 独立开源开发者

Front-End Jobs by Framework



2.2三大前端框架：VUE、REACT、ANGULAR (2)

■一些比较与资源

- https://blog.csdn.net/weixin_45869811/article/details/127627143
- https://blog.csdn.net/qq_40436854/article/details/103595280

```
## Github
```

```
### React
```

```
https://github.com/facebook/react
```

```
### Angular
```

```
https://github.com/angular/angular
```

```
### Vue
```

```
https://github.com/vuejs/vue
```


2.3 VUE的安装方法

■ HX与Uniapp已经包含了对Vue开发所需所有设置

■ Node.js [官网 Node.js \(nodejs.org\)](https://nodejs.org)

- Node.js® is an open-source, cross-platform JavaScript runtime environment.
- 下载（30M） 安装
- 包含npm



ABOUT NPM

- npm is the world's largest **software registry**. Open source developers from every continent use npm **to share and borrow packages**, and many organizations use npm to manage private development as well.
- npm consists of three distinct components:
 - **the website**: to discover packages, set up profiles, and manage other aspects of your npm experience. For example, you can set up organizations to manage access to public or private packages.
 - the Command Line Interface (**CLI**): runs from a terminal, and is how most developers interact with npm.
 - **the registry**: is a large public database of JavaScript software and the meta-information surrounding it.

■ Npm工具资源库在国外，传输速度慢，建议使用国内镜像

- 如淘宝的cnpm, 并使用cnpm代替npm进行后续安装。
- `npm install -g cnpm --registry=https://registry.npm.taobao.org`
- CLI翻译为命令行界面，俗称脚手架。Vite（作者是尤雨溪）是一个轻量级的、快速的构建工具，对 Vue SFC 提供第一优先级支持。

- node.js版本 node -v
- npm init vue@latest 名称不要出现大写
- 创建的最简单的一个vue项目

> Su > 教学 学科实践 (一) > code > vue_base >	
名称	修改日期
^	
📁 .vscode	2023/7,
📁 public	2023/7,
📁 src	2023/7,
📄 .gitignore	2023/7,
🌐 index.html	2023/7,
📄 package.json	2023/7,
📄 README.md	2023/7,
📄 vite.config.js	2023/7,

> Su > 教学 学科实践 (一) > code > vue_base > src	
名称	修改日期
^	
📁 assets	2023/7/16 9
📁 components	2023/7/16 9
📄 App.vue	2023/7/16 8
📄 main.js	2023/7/16 8

VUE(VSCODE下) 的目录结构

- .vscode目录, 属于开发工具的, 不属于vue, 对应的是.hbuilderX
- Node+modules: 执行npm install的产物
- Public下的.icon文件, 是运行时浏览器左上角的图标所用
- Git 团队开发协同文件
- Vite是vue的一个构建工具
- Json文件很重要

1	.vscode	--- VSCode工具的配置文件夹
2	node_modules	--- Vue项目的运行依赖文件夹
3	public	--- 资源文件夹 (浏览器图标)
4	src	--- 源码文件夹
5	.gitignore	--- git忽略文件
6	index.html	--- 入口HTML文件
7	package.json	--- 信息描述文件
8	README.md	--- 注释文件
9	vite.config.js	--- Vue配置文件

以CDN形式引入VUE

- CDN: Content Delivery Network
- CDN引入Vue.js的方式不适合在生产环境中使用, 因为CDN的可靠性和性能可能无法保证。
- 在生产环境中, 建议使用npm或其他包管理工具来安装和管理Vue.js。

```
<!DOCTYPE html>
<html>
  <head>
    <title>Vue.js CDN Example</title>
    <script src="https://cdn.jsdelivr.net/npm/vue"></script>
  </head>
  <body>
    <div id="app">
      {{ message }}
    </div>
    <script>
      var app = new Vue({
        el: '#app',
        data: {
          message: 'Hello, Vue!'
        }
      })
    </script>
  </body>
</html>
```

VUE的SFC（单文件组件）

- SFC (Single-File Component: 即 *.vue 文件) 是一种特殊的文件格式，它将一个 Vue 组件的模板、逻辑与样式封装在单个文件中。
- SFC是HTML、CSS 和 JS 三种语言经典组合的自然延伸。
<template>、<script> 和 <style> 三个顶级块在同一个文件中封装、组合了组件的视图、逻辑和样式

三、VUE开发与传统前端开发的不同

from 白话uni-app(<https://uniapp.dcloud.net.cn/tutorial/vue3-basics.html#%E7%99%BD%E8%AF%9Duni-app>)

- **文件类型变化：** .vue文件，开发是vue，经过编译后，运行时变成了js（有编译器、运行时的概念）
- **文件内代码架构的变化：** SFC由 template （用于写tag组件）、script 和 style 三个一级节点组成。

三、VUE开发与传统前端开发的不同(2)

■外部文件引用方式变化

- `import` 引入外部的js模块(注意不是文件)或css;
- 把js模块 `require`进来,变成了对象来使用。把function 封装为模块(module)的方法并导出(exports)。只有被导出的方法和属性才能被外部调用。
- 全局样式可以在app.vue中引入。
- Vue支持组件(其本质是:封装的包括界面、js、样式的库)导入。

三、VUE开发与传统前端开发的不同(3)

■ 组件/标签的变化

- 以前是html标签（属于浏览器内置的东西），比如 `<div>`，现在是小程序组件，比如 `<view>`。
- 组件可以自由扩展。例如把一段js封装成函数或模块，或者把一个ui控件封装成一个组件。
- uni-app 参考小程序规范，提供了一批内置组件

三、VUE开发与传统前端开发的不同(4)

■js的变化

- script里默认有export default，在里面写data、事件和method
- 写在 export default { 前面的变量，是页面内部的全局变量，可以在各种方法里使用。
- export default {} 里是一个大json，data、生命周期、method都需要用逗号分隔。
- data -> return 里，编写可以绑定在页面template模板里的变量，页面组件的text里绑定data数据使用{{}}，比如下面例子中的textvalue。而下面的globalvar就不能在模板里绑定使用。在HBuilderX中，敲vdata代码块，可以快速生成data的代码结构。
- 页面的生命周期/事件，（如onLoad），和data同级。
- 模板里要调用的方法，都需要写在methods下面。每个方法也需要用逗号分隔。不需要再使用function声明，只要写在methods下的函数，都可以在template里调用。

四、UNIAPP页面

1. 解读pages.json
2. 解读index.vue
3. 丰富 index.vue
4. 添加error.vue并修改pages.json

■ 4.1 解读 ages.json

```
{  
  "pages": [],  
  "globalStyle": {},  
  "uniIdRouter": {}  
}
```

- uniIdRouter 是一个运行在前端的、对前端页面访问权限路由进行控制的方案。在uniCloud中直接有提供好的uniIdRouter解决方案，我们只需要在项目中进行简单配置即可达到使用目的。

```
pages.json  
1 {  
2   "pages": [ //pages数组中第一项表示应用启动页, 参考  
3     {  
4       "path": "pages/index/index",  
5       "style": {  
6         "navigationBarTitleText": "uni-app"  
7       }  
8     }  
9   ],  
10  "globalStyle": {  
11    "navigationBarTextStyle": "black",  
12    "navigationBarTitleText": "uni-app",  
13    "navigationBarBackgroundColor": "#F8F8F8",  
14    "backgroundColor": "#F8F8F8"  
15  },  
16  "uniIdRouter": {}  
17 }  
18
```

4.2 解读index.vue <template>

```
1 <template>
2   <view class="content">  仅有一个一级view
3     <image class="logo" src="/static/logo.png"></image>
4     <view class="text-area">
5       <text class="title">{{title}}</text>
6     </view>
7   </view>
8 </template>
```

文本插值

■ 解读index.vue <script>

```
10 <script>
```

```
11   export default {
```

```
12     data() {
```

```
13       return {
```

```
14         title: 'Hello'
```

```
15       }
```

```
16     },
```

```
17     onLoad() {
```

```
18     },
```

```
19     methods: {
```

```
20     }
```

```
21   }
```

```
22 </script>
```

是个函数，返回一个对象

一个函数，属于页面声明周期

方法集，是一个对象，可以包含很多函数

■ 丰富 4.3index.vue

```
onShow(){
    console.log('index page showed')
},
onLoad() {
    console.log('index page load')
},
onPullDownRefresh(){
    console.log('页面下拉刷新记录')
},
```



```
1 <template>
2   <view> <text>当前页面不存在! </text> </view>
3 </template>
```

```
4 <script>
```

```
5   export default {
```

```
6     data() {
```

```
7       return {
```

```
8     }
```

```
9   }
10   onLoad() {
```

```
11     let timer=setTimeout(()=>{
```

```
12       clearTimeout(timer)
```

```
13       uni.navigateTo({
```

```
14         url:'/pages/index/index',
```

```
15       })
```

```
16     },2000)
```

```
17   },
```

```
18   methods: {
```

```
19   }
```

```
20 }
```

```
21 </script>
```

4.4 添加404.vue,并修改pages.json

```
"pages": [ //pages数组中第一项表示应用启动页, 参
{
  "path": "pages/error/404",
  "style": {
    "navigationBarTitleText": "uni-app-404"
  }
}
```

404.vue

```
1 <template>
2   <view>
3     <text>当前页面不存在! </text>
4   </view>
5 </template>
6
7 <script>
8   export default {
9     data() {
10       return {
11
12       }
13     },
14     onLoad() {
15       let timer=setTimeout(()=>{
16         clearTimeout(timer)
17         uni.navigateTo({
18           url: '/pages/index/index',
19         })
20       },2000)
21     },
22     methods: {
23
24     }
25   }
26 </script>
```

404.vue

App.vue ×

```
1 <script>
2   export default {
3     onLaunch: function() {
4       console.log('App Launch')
5     },
6     onShow: function() {
7       console.log('App Show')
8     },
9     onHide: function() {
10      console.log('App Hide')
11    },
12    onPageNotFound(){
13      uni.navigateTo({
14        url: '/pages/404/404'
15      })
16    }
17  }
18 </script>
19
20 <style>
21   /*每个页面公共css */
22 </style>
```

404.vue

index.html

index.vue ×

```
7
8   </view>
9 </template>
10
11 <script>
12   export default {
13     data() {
14       return {
15         title: 'Hello'
16       }
17     },
18     onShow(){
19       console.log('index page showed')
20     },
21     onLoad() {
22       console.log('index page load')
23     },
24     onPullDownRefresh(){
25       console.log('页面下拉刷新记录')
26     },
27     methods: {
28
29     }
30   }
31 </script>
```

基础

你好，世界

处理用户输入

Attribute 绑定

条件与循环

表单绑定

简单组件

本章作业

本周不用交作业，与后续作业一起交

作业1：<https://cn.vuejs.org/examples/#hello-world> 对该页面进行改造，完成右边红框中的示例的阅读与运行，并截图。

作业2 阅读 页面简介

(<https://uniapp.dcloud.net.cn/tutorial/page.html#%E9%A1%B5%E9%9D%A2%E7%AE%80%E4%BB%8B>)，理解“页面生命周期”：理解“应用生命周期”

作业3

- Pages目录下添加页面404，（注意，在pages.json的变化）
添加 组件text
- 在app.vue中添加 onPageNotFound(), 导航到url 404
- 在微信开发者工具中，“添加编译模式”，使得启动页面指向不存在的页面，从而激发onPageNotFound()
- 添加404的onLoad ()，使得404页面停留2秒后，跳回index页面
- 在pages.json 文件中，对index “页面，添加style属性
“enablePullDownRefresh” : true。回到在index页面，调用onShow(), onLoad(), onPullDownRefresh(), 体会页面生命周期。