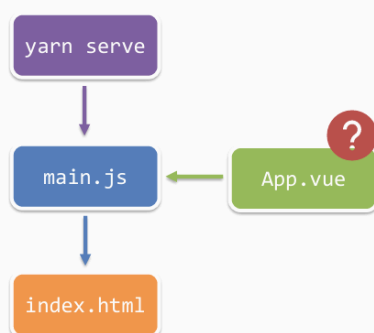


VUE 目录作用

脚手架目录文件介绍 & 项目运行流程

VUE-DEMO	
├─node_modules	第三包文件夹
├─public	放html文件的地方
│ ├─favicon.ico	网站图标
│ └─index.html	index.html 模板文件 ③
├─src	源代码目录 → 以后写代码的文件夹
│ ├─assets	静态资源目录 → 存放图片、字体等
│ ├─components	组件目录 → 存放通用组件
│ ├─App.vue	App根组件 → 项目运行看到的内容就在这里编写 ②
│ └─main.js	入口文件 → 打包或运行，第一个执行的文件 ①
├─.gitignore	git忽视文件
├─babel.config.js	babel配置文件
├─jsconfig.json	js配置文件
├─package.json	项目配置文件 → 包含项目名、版本号、scripts、依赖包等
├─README.md	项目说明文档
├─vue.config.js	vue-cli配置文件
└─yarn.lock	yarn锁文件，由yarn自动生成的，锁定安装版本

脚手架目录文件介绍 & 项目运行流程



main.js 核心代码

1. 导入 Vue
`import Vue from 'vue'`
2. 导入 App.vue
`import App from './App.vue'`
3. 实例化 Vue，将 App.vue 渲染到 index.html 容器中
`new Vue({
 render: h => h(App),
}).$mount('#app')`

```
src > JS main.js > ...
1  // 文件核心作用：导入VUE组件，基于app.vue文件创建Vue实例
2  // 导入 VUE核心包
3  import Vue from 'vue'
4  // 导入 APP.vue 根组件
5  import App from './App.vue'
6
7  // 提示：当前处于什么环境
8  Vue.config.productionTip = false
9
10 // vue实例化 |
11 new Vue({
12   render: h => h(App),
13 }).$mount('#app')
14
```

组件化开发

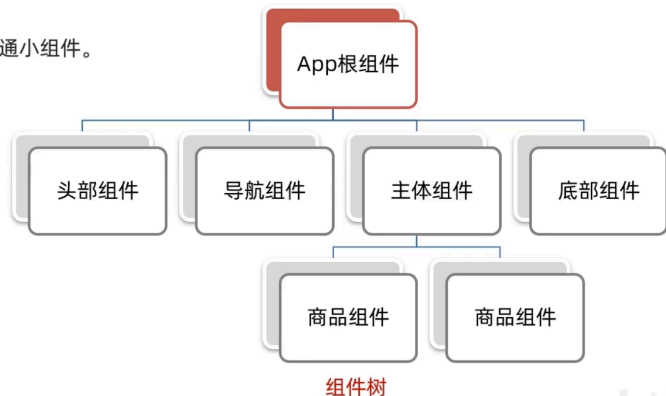
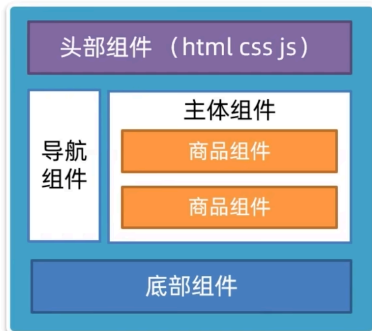
组件化开发 & 根组件

① **组件化**：一个页面可以拆分成一个个组件，每个组件有着自己独立的结构、样式、行为。

好处：便于**维护**，利于**复用** → 提升**开发效率**。

组件分类：普通组件、根组件。

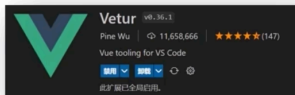
② **根组件**：整个应用最上层的组件，包裹所有普通小组件。



组件树

App.vue 文件（单文件组件）的三个组成部分

1. 语法高亮插件：

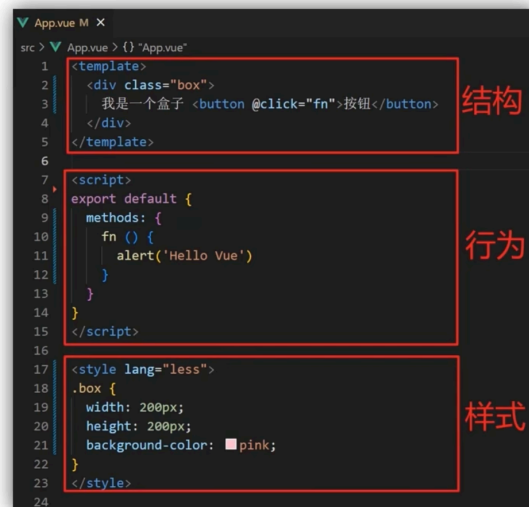


2. 三部分组成：

- ◆ **template**：结构（有且只能一个根元素）
- ◆ **script**：js逻辑
- ◆ **style**：样式（可支持less，需要装包）

3. 让组件支持 less

- (1) **style**标签，**lang="less"** 开启less功能
- (2) 装包：**yarn add less less-loader**



局部组件开发

普通组件的注册使用

组件注册的两种方式：

1. 局部注册：只能在注册的组件内使用

- ① 创建 .vue 文件 (三个组成部分)
- ② 在使用的组件内导入并注册

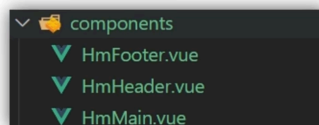
2. 全局注册：所有组件内都能使用

使用：

- ◆ 当成 html 标签使用 `<组件名></组件名>`

注意：

- ◆ 组件名规范 → 大驼峰命名法，如：HmHeader



```
// 导入需要注册的组件
import 组件对象 from '.vue文件路径'
import HmHeader from './components/HmHeader'

export default {
  // 局部注册
  components: {
    '组件名': 组件对象,
    HmHeader: HmHeader
  }
}
```

全局注册组件

```
src > JS main.js > ...
1  // 文件核心作用：导入VUE组件，基于app.vue文件创建Vue实例
2  // 导入 VUE核心包
3  import Vue from 'vue'
4  // 导入 APP.vue 根组件
5  import App from './App.vue'
6
7  // 导入全局组件
8  import LinButton from './components/LinButton.vue'
9
10 // 提示：当前处于什么环境
11 Vue.config.productionTip = false
12
13 // 全局注册注册
14 Vue.component('LinButton', LinButton)
15 // vue实例化
16 new Vue({
17   render: h => h(App),
18 }).$mount('#app')
19
```

Scoped 样式只在本组件中生效



组件的样式冲突 scoped

默认情况：写在组件中的样式会 **全局生效** → 因此很容易造成多个组件之间的样式冲突问题。

1. **全局样式**：默认组件中的样式会作用到全局
2. **局部样式**：可以给组件加上 **scoped** 属性, 可以让样式只作用于当前组件

Scoped 属性原理

```
src > components > BaseOne.vue > {} "BaseOne.vue" > style
18 2. 加上scoped属性的style样式, 只会作用于当前组件 → 局部样式
19
20 组件应该有着自己独立的样式, 推荐加上scoped(原理)
21 -----
22 scoped原理:
23 1. 给当前组件模板的所有元素, 都会被添加上一个自定义属性
24    data-v-hash值
25    data-v-5f6a9d56 区分不同的组件
26 2. css选择器后面, 被自动处理, 添加上了属性选择器
27    div[data-v-5f6a9d56]
28 */
29 div {
30   border: 3px solid blue;
31   margin: 30px;
```

组件中 Data是一个函数

保证每个组件实例维护一份独立数据

data 是一个函数

一个组件的 **data** 选项必须是一个**函数**。→ 保证每个组件实例, 维护**独立**的一份数据对象。

每次创建新的组件实例, 都会新执行一次 data 函数, 得到一个**新对象**。

