03.组件化开发

学习要点：

1.开发思维

2.小实例演示

本节课我们来开始了解 Vue-cli 组件化开发的思路，然后演示小实例。

一. 开发思维

1. 在使用脚手架开发时，使用的是组件化的开发模式，也就是.vue 文件；
2. 一个.vue 文件就是一个组件，里面基本包含：html、js 和 css 三个部分；
3. 这种方式，分离了每个组件的关联，提高了内聚，所以叫：组件化开发；
4. 开发完毕后，再进行打包编译成常规模式即可运行；
5. 默认创建的项目，src 包含两个文件：App.vue 和 main.js；
6. App.vue 是根组件，而 main.js 是入口文件，注释如下：

```
// 引入 Vue 框架
import { createApp } from 'vue'
// 引入 App.vue 根组件
import App from './App.vue'

// 挂载
createApp(App).mount('#app')
```

7. 下面，我们看下 App.vue 中，引入了一个测试组件：HelloWorld，我们先注释掉；
8. 取消掉测试组件后，查看页面元素或源代码，你会发现有一个基本的 html 结构；
9. 这个基本的 html 结构是脚手架默认提供的模板 public/index.html；
10. 而当我们引入这个测试组件后，在#app 里就添加了组件的全部内容；

二. 小实例演示

1. 我们模仿脚手架给的测试组件，自行创建两个组件，并引入到根组件；

```
<template>
    <img alt="Vue logo" src="./assets/logo.png">
    <Sidebar></Sidebar>
    <Footer></Footer>
</template>

<script>
    import Sidebar from "./components/Sidebar"
    import Footer from "./components/Footer"

    export default {
        name: 'App',
        components: {
```

```
            //HelloWorld
            Sidebar,
            Footer
        }
    }
</script>
```

PS：子组件 Sidebar 和 Footer 会自动生成基本格式，任意输入内容即可；