

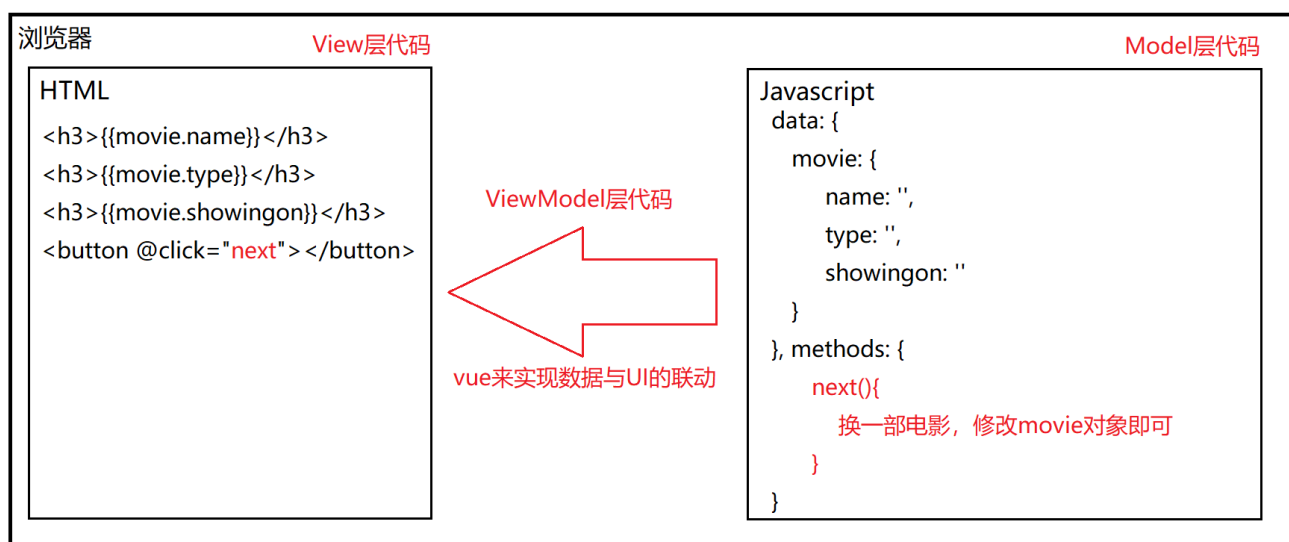
# Vue DAY02

## MVVM架构

M - Model层 业务逻辑层 主要用于管理数据，实现业务

V - View层 视图层 主要用于将数据显示在页面上

VM - ViewModel层 视图模型层 主要用于业务逻辑与视图的联动（Vue来做）



## 基于脚手架环境开发vue项目

开发vue项目有两种模式：

1. 基于vue.js，在html页面中引入vue.js，让vue.js管理div#app元素。
2. 基于脚手架环境：通过vue脚手架环境可以方便的创建一个通用的vue项目框架模板，在此基础上开发vue项目更加快捷，适合工程化开发。

## VueCLI脚手架开发环境

<https://cli.vuejs.org/>

### 安装脚手架

```
# 设置npm源地址
npm set registry https://registry.npmirror.com

# 安装脚手架
npm install -g @vue/cli
```

安装配置VueCLI，通过VueCLI创建项目的过程，详情查看：

ftp目录/VueBasic/day02/resource/VueCLI\_installation\_guide.pdf

找一个干净地方，新建项目包：

```
# day02/demo/  
vue create vue-project
```

一旦项目创建完毕后，即可通过命令，启动脚手架项目（端口为8080的服务），提供网站的访问。

```
npm run serve
```

打开浏览器：<http://localhost:8080/>

该项目的node\_modules目录是不会上传远程仓库的，所以从仓库下载的项目包是很小的，如果需要运行，则应该在package.json所在目录执行命令，安装所需要的依赖模块：

```
npm install
```

脚手架运行时涉及到的文件：

```
public/index.html  
src/main.js  
src/App.vue  
src/views/HomeView.vue (先放下)  
src/views/AboutView.vue (先放下)
```

**脚手架的运行过程：**当执行 `npm run serve` 时，将会编译整个项目。启动8080服务来提供网站的访问。当访问地址：<http://localhost:8080> 时，将会看到 `public/index.html`，并且在该页面中将自动加载 `main.js` 中的代码，创建vue对象，由该vue对象来管理页面中的 `#app`。

```
new Vue({  
  router,  
  store,  
  render: h => h(App)  
}).$mount('#app')
```

初始化的状态下，默认将会把App.vue组件（`Component`）中的内容渲染到#app中，从而看到页面效果。

**需求：设计一个页面，使用脚手架实现。**

访问：<http://localhost:8080/login>，可以看到一个登录页面：包含账号、密码、登录按钮。

# 脚手架路由系统（VueRouter）的设计与使用

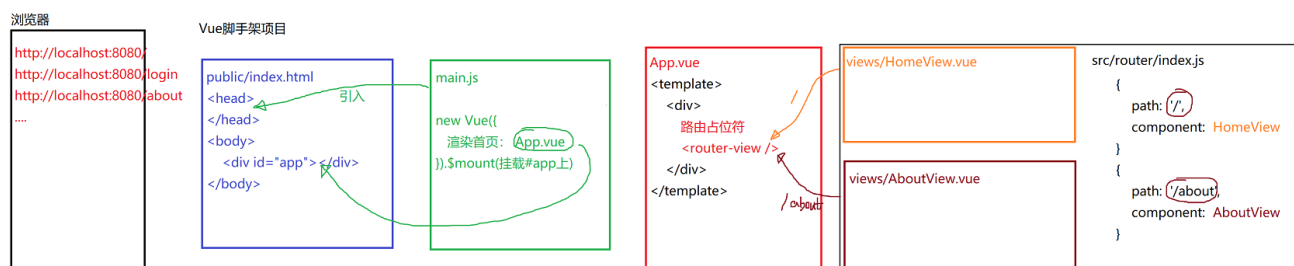
脚手架的路由系统的功能是通过浏览器地址栏的地址来动态显示页面内容。

当访问：/login时，可以看到Login页面。

当访问：/时，可以看到HomeView页面。

当访问：/about时，可以看到AboutView页面。

.....



在路由系统中有个非常重要的组件：`<router-view />`，它可以根据请求路径的不同，动态的在该位置显示相应组件内容。具体如何显示得参照 `src/router/index.js` 的配置：

```
const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/about',
    name: 'about',
    component: () => import('../views/AboutView.vue')
  }
]
```

思考题：尝试实现刚才的登录页面的需求。

---

作业：建10个项目