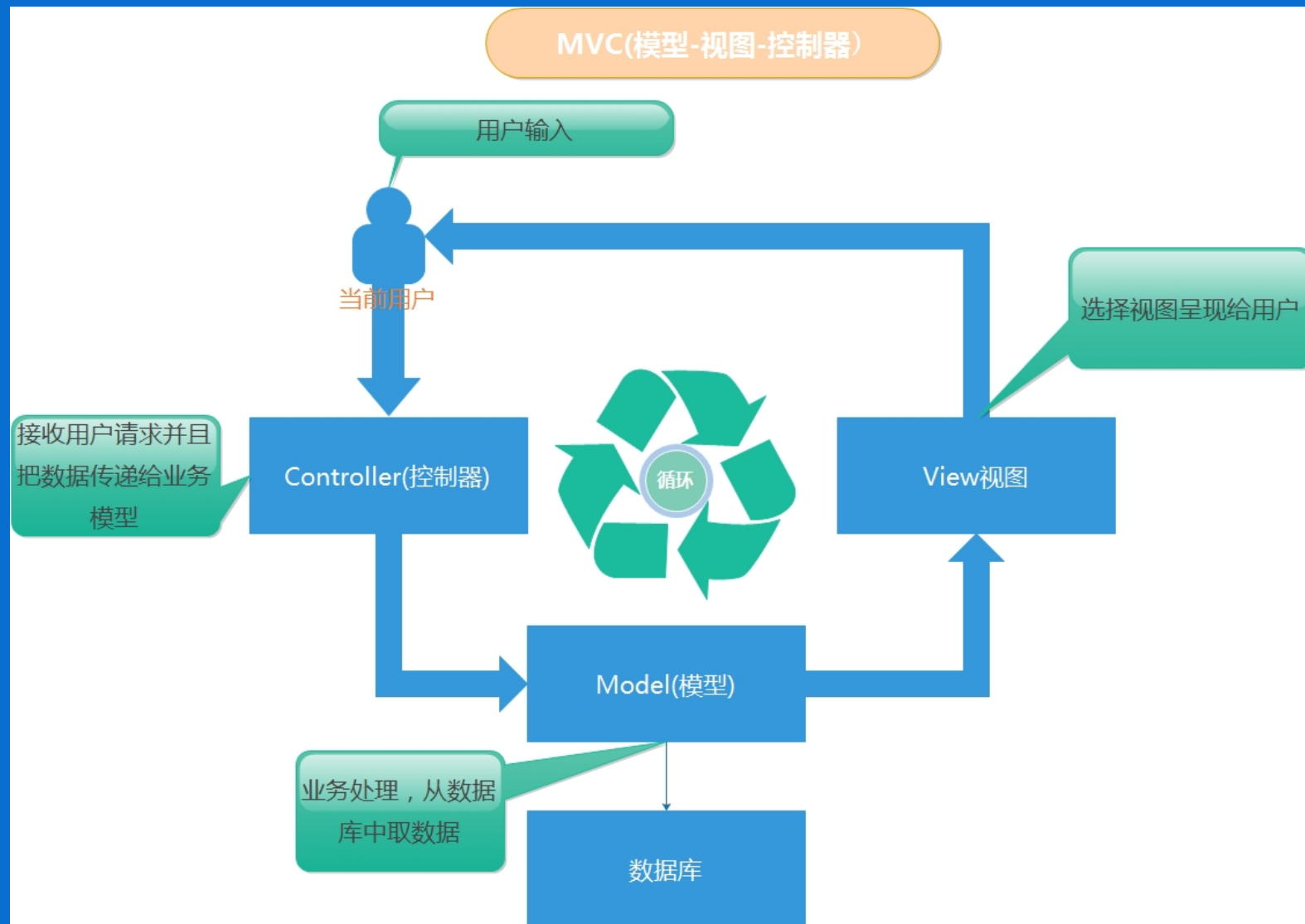


MVC

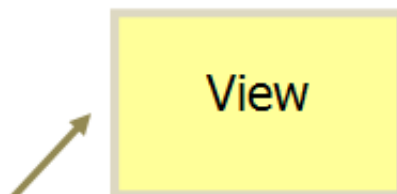
分类	用途
Model	模型表示 数据 的部分
View	视图是用户看到并与之交互的 界面
Controller	控制器是应用程序中处理用户 交互 的部分。通常控制器负责从视图读取数据，控制用户输入，并向模型发送数据。

- MVC 是一种使用 MVC (Model View Controller **模型 - 视图 - 控制器**) 设计创建 Web 应用程序的模式
- MVC **分层** 同时也简化了 **分组** 开发。不同的开发人员可同时开发视图、控制器逻辑和业务逻辑。

MVC流程



的变化。这些可以通过 **View** 来发送和渲染，由HTML来展示你的代码。



使用MVVM模式好处

1. **低耦合**：View可以独立于Model变化和修改，一个ViewModel可以绑定到不同的View上
2. **可重用性**：可以把一些视图的逻辑放在ViewModel里面，让很多View重用这段视图逻辑。
3. **独立开发**：开发人员可以专注与业务逻辑和数据的开发(ViewModel)，设计人员可以专注于界面(View)的设计。
4. **可测试性**：可以针对ViewModel来对界面(View)进行测试。

指令系统

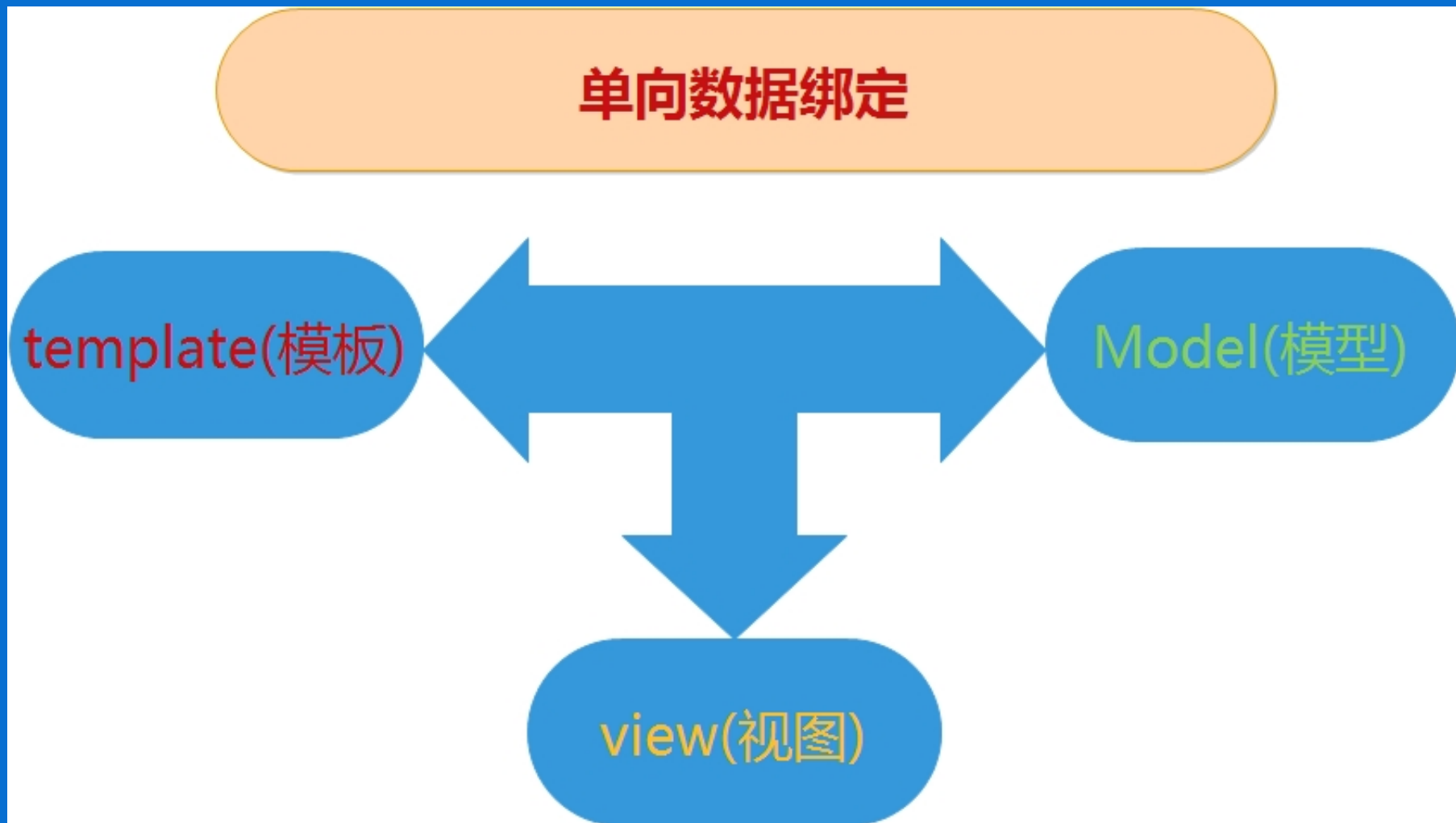
- 指令指的是当关联的HTML结构进入 **编译** 阶段时应该执行的操作
- 当浏览器启动、开始解析HTML时，DOM元素上的指令属性就会跟其他属性一样被解析，也就是说当一个Angular应用启动
 1. Angular编译器就会遍历DOM树来解析HTML， **寻找** 这些指令属性函数
 2. 在一个DOM元素上找到一个或多个这样的指令属性函数
 3. 它们就会被 **收集** 起来、 **排序** ，然后按照优先级顺序被执行。
- **组件型** 指令主要是为了将 **复杂的视图分离** ，实现高内聚低耦合，是自封装和内聚性的整体，包含了视图和交互逻辑,比如登陆框
- **装饰型** 指令是为DOM添加行为或使之具有某些 **能力** ，如可拖动、比如添加样式等

依赖注入

只需要声明要使用的对象，就可以自动得到，不需要自己去创建



单向数据绑定

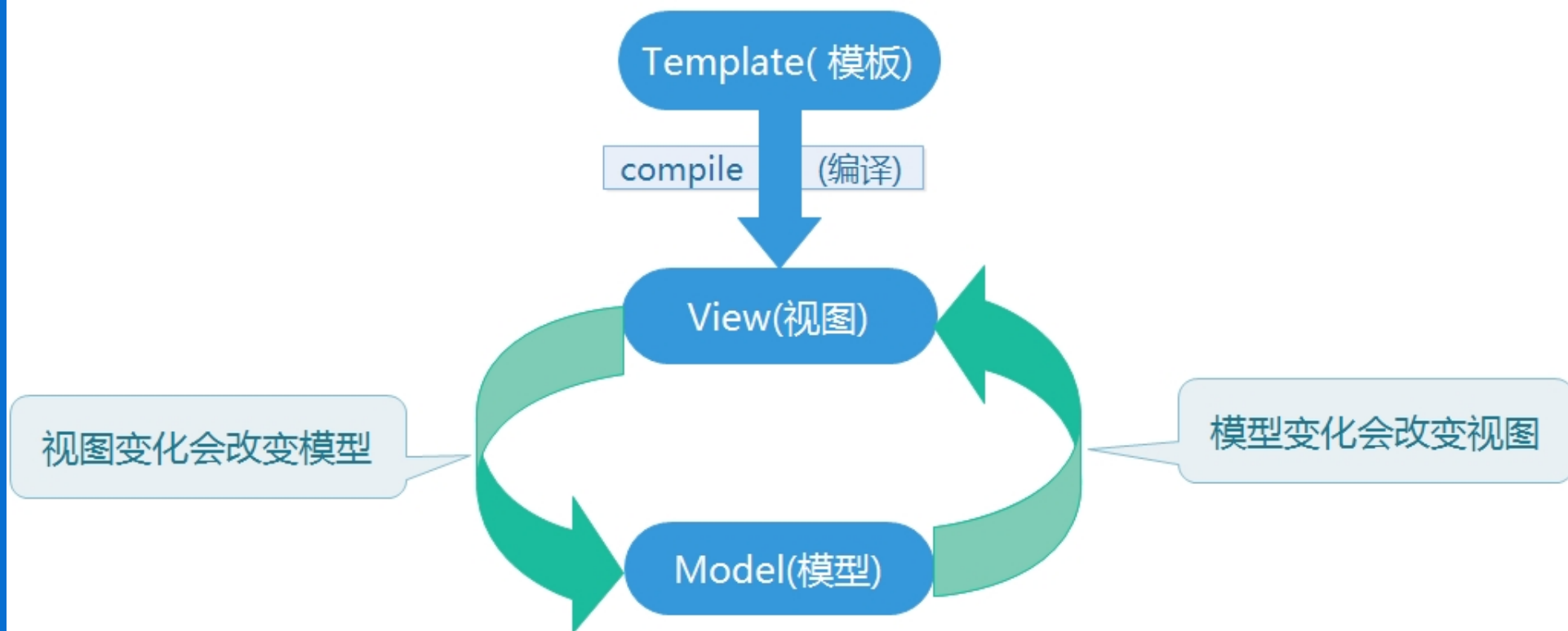


双向数据绑定

- 当 `View` 中有任何数据发生了变化，那么这个变化也会自动地反馈到 `scope` 的数据上，`scope` 模型会自动地更新
- 当 `scope` 模型发生变化时，`view` 中的数据也会更新到最新的值

双向数据绑定

双向数据绑定



过滤器

- filter是用来进行数据格式化的

过滤器	作用
lowercase	转小写
uppercase	转大写