1. 特性：双向数据绑定、声明式依赖注入、解耦应用逻辑，数据模型和视图、完善的页面指令、定制表单验证、Ajax封装
   1. 数据绑定：数据从一个地方转移到另一个地方
   2. 双向数据绑定：数据可以从View（视图层）流向Model，也可以从Model流向View
      1. ng-model是双向的、｛｛｝｝是单向的
2. angular是JS结构化框架，主体不再是DOM，而是页面中的动态数据
3. 应用：构建单页面Web应用（SPA）或Web App应用
4. v1.x语法：
   1. <标签body ng-app=“模块对象名”>：
      1. ng-xxx：指令，ng代表angular
         1. ng-app 指定angular核心管理当前标签所包含的整个区域，并自动创建$rootScope根作用域对象
         2. ng-init = “变量名=‘初始值’” 初始化时执行表达式
         3. ng-model=“变量名” 将元素的value特性与当前作用域对象上的指定属性绑定
         4. ng-click=“回调函数名” 点击时，调用作用域对象的方法
         5. ng-bind=“表达式” 解决先加载html标签导致的表达式先显示，angular再解析的问题
         6. ng-repeat=“变量名 in 数组名” 遍历数组显示数据，数组有几个元素就会产生几个新的作用域
            1. 创建一个作用域，把标签内部的内容归入该作用域并加入到HTML文档中，重复上述步骤直至遍历完成
            2. 标签内部可以用｛｛变量名｝｝使用变量。且自带以下属性：$index、$first、$last、$middle、$odd、$even
         7. ng-show / hide=“布尔或变量” 当值为true时显示/隐藏
   2. <p>{{表达式}}</p>：返回表达式的值，可以放在任何需要值的地方，从当前作用域对象上取
      1. 表达式中undefined、Infinity、NaN、null解析为空串“”
   3. 依赖对象：完成某个特定的功能需要某个对象才能实现，这个对象就是依赖对象
   4. 依赖注入：依赖的对象以形参的形式被注入进来使用
   5. 作用域对象：一个js实例对象，它的属性和方法与页面中的指令或表达式相关联
   6. 控制器对象：用来控制angularJS应用数据的js实例对象
      1. <标签 ng-controller=“构造函数名”> : 创建控制器对象（老版本）
         1. 在js代码中声明同名函数，其形参必须是$scope（依赖对象），可以在函数内对$scope进行属性添加修改，即对作用域对象修改
         2. 指定控制器构造函数, Angular会自动new此函数创建控制器对象。同时Angular还有创建一个新的域对象$scope, 它是$rootScope的子对象。在控制器函数中声明$scope形参, Angular会自动将$scope传入
   7. 模块对象：
      1. let 模块对象名= angular.module（‘模块对象名’，[]） 创建模块对象，数组内写需要的模块
      2. 方法：
         1. .controller（‘控制器名’，[‘$scope’，function（$scope）｛｝]） 创建控制器对象，返回模块对象（链式调用），显式声明依赖注入