# Angular 模板

# 模板基础

## 什么是模板

在 Angular 中，组件扮演着控制器或视图模型的角色，模板则扮演视图的角色。模板是一种自定义的标准化页面，通过模板和模板中的数据结合，可以生产各种各样的网页。在Angular中，模板的默认语言是HTML，几乎所有的HTML语法在模板中都是适用的，但<script>标签是被禁止的，主要是为了防止JavaScript脚本注入攻击（XXS）。同时一些HTML元素在模板中并不起什么作用，比如<html>、<body>、<base>等。除此之外，Angular中可以通过组件和指令对模板的HTML元素进行扩展，这些扩展将以新的元素或属性的形式出现。

## 模板表达式

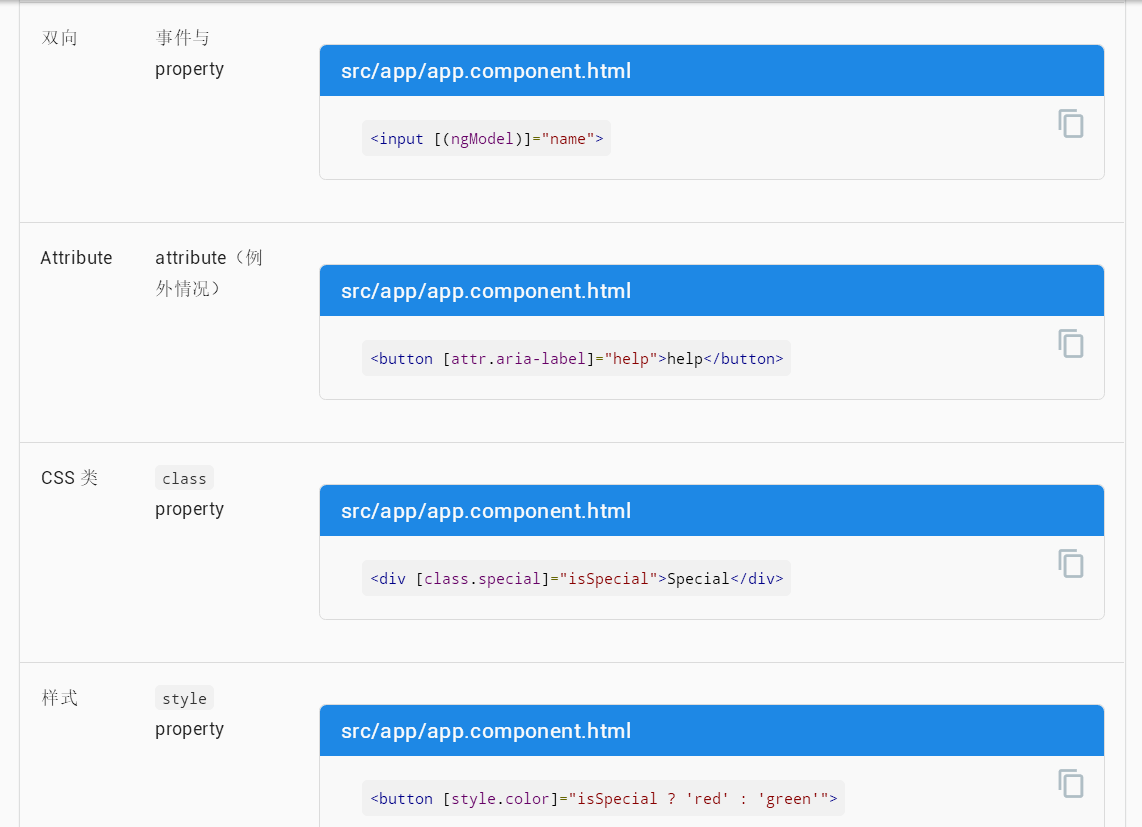
模板表达式类似于JavaScript语言的表达式，绝大部分的JavaScript表达式都是合法的模板表达式。模板表达式应用在插值语法的双大括号中和属性绑定 “=”右侧的引号中。Angular会执行这个表达式并将值分配给一个绑定目标的属性，这个绑定目标可能是一个HTML元素、组件或者指令

注意不能为一下几种情况：

1. 带有new运算符的表达式。
2. 赋值表达式（=、+=、-=）等。
3. 带有;或者,的链式表达式。
4. 带有自增和自减操作（++/--）的表达式
5. 不支持位运算符（| 和 &）

请问三元运行符能不能用？ 可以！

## 语法大全



# 数据绑定

Angular中不论是从组件控制器中的属性绑定到模板上，还是从模板中的事件反映到控制器中，它们都默认是单向的。

## 插值表达式

语法：{{ }}，也是单向绑定

## 事件绑定

数据从模板流向组件类。在事件绑定中，Angular通过监听用户操作事件，比如键盘事件、鼠标事件等来执行其对应绑定的方法。事件绑定的语法是由“=”左侧小括号内的目标事件和“=”右侧引号中的目标语句组成。示例 (click)=”onClick”

类型：

1. 绑定自定义事件。 场景：子组件@Output发射出来的事件。
2. DOM原生事件。 事件一览表：

<https://developer.mozilla.org/zh-CN/docs/Web/Events>

## 属性绑定

在学习属性绑定前先来了解一对非常重要的概念，即DOM对象属性（Property）与HTML标签特性（Attribute）。在英语中Property和Attribute都可以译为‘属性’，名字虽然相同，但在模板中意义却大有不同。

两者的区别和联系还体现在以下两个方面：

1. 大多数情况下，DOM对象属性与HTML标签特性并不是一一对应的，但有少量属性即是DOM对象属性又是HTML标签特性，如id、title、class 等
2. 通常HTML标签特性代表初始值，初始化以后就不发生改变，而DOM元素对象属性代表当前值，其默认为初始值但是会随着改变。

实际的使用：大多数情况我们都是绑定DOM Property属性，而有些特例你只能绑定attr （colspan，我们table中的标签属性）[attr.colspan]

## 双向数据绑定

我们经常需要显示数据属性，并在用户作出更改时更新该属性。

在元素层面上，既要设置元素属性，又要监听元素事件变化。

Angular 为此提供一种特殊的双向数据绑定语法：[(x)]。 [(x)]语法结合了属性绑定的方括号[x]和事件绑定的圆括号(x)。

原理：

一个 @Input xxx 加上 @Output xxxChange，构成 [( xxx )];

## 模板引用变量

模板引用变量通常用来引用模板中的某个DOM元素，它还可以引用Angular组件或指令或Web Component。示例：使用井号 (#) 来声明引用变量。也可以使用ref- 前缀代替。

## 模板表达式操作符

1. 管道操作符 ( | )
2. 安全导航操作符 ( ?. ) 和空属性路径
3. 非空断言操作符（!） // 需要特定的场景

# 管道

在Angular中，管道（Pipes）可以按照开发者指定的规则将模板内的数据进行转换。如日期类型（Date），默认会显示为Mon Sep 18 2017 17:16:11 GMT+0800 (中国标准时间) 这样的形式，但不如Sep 18,2017这样的形式直观，此时可以借助管道来实现这种转换效果。

## 管道介绍

使用管道，需要用管道操作符 | 来连接模板表达式中左边的输入数据和右边的管道，示例：

{{ birthday | date }}

以上例子中，通过管道操作符右边的date管道来实现日期数据的格式转换，date管道是Angular的内置管道，存放在Common-Module里。

## 管道参数

管道可以使用参数，示例：

{{ birthday | date:”y/MM/dd” }}

通过传入的参数来输出不同格式的数据

y 代表年， M代表月份，d代表天， H代表小时，m代表分钟，s代表秒

## 链式管道

一个模板表达式可以连续使用多个管道进行不同的处理，这就是链式管道。使用链式管道可以展示更丰富的数据格式，语法格式如下所示:

{{ birthday | date | uppercase }}

## 常用内置管道

### DatePipe

用来格式化一个日期数据，具体用法参照上面示例

### JsonPipe

通过JSON.stringify()来将输入数据对象转换成对象字符串，主要用于开发调试。示例：{{ jsonObject | json }}

### DecimalPipe

用于对数值的整数与小数部分按照指定规则进行格式化，这种格式化方式也称为本地格式化处理，示例：

expression | number[: {minIntegerDigits}.{minFractionDigits}-{maxFractionDigits}]

内置管道全家福：

<https://www.angular.cn/api?type=pipe>

### 自定义管道

虽然Angular提供了许多内置的管道，但是数据转换涉及各种各样的格式，内置管道显然无法满足全部需求，因此我们需要使用Angular提供的自定义管道功能来实现更多的需求。

命令： ng g p xxx

步骤：

1. 新建一个pipe， ng g p sexreform
2. Angular就会在根模块中进行处理。
3. 实现一个transform的方法，第一个参数是被管道的值，第二个参数是我们的管道参数（可选）。 需要返回被处理完之后的值

注意：

1、@Pipe 装饰器装饰的。

2、name元数据的值决定了管道的使用名称。

3、管道并不会改变原始数据的值，只是进行了一个格式化

### 管道分类

纯管道：Angular只有在它检测到输入值发生了纯变更时才会执行纯管道。 纯变更是指对原始类型值(String、Number、Boolean、Symbol)的更改， 或者对对象引用(Date、Array、Function、Object)的更改。

非纯管道：Angular会在每个组件的变更检测周期中执行非纯管道。 非纯管道可能会被调用很多次，和每个按键或每次鼠标移动一样频繁。示例：| async（Rxjs中会提到）

# 查漏补缺&答疑&半程总结

## @contentChild

我们讲过 @viewChild，父组件在控制器中获取子组件的实例。

@ contentChild，在自组件中使用的。

## :host

<https://www.angular.cn/guide/component-styles#host-选择器>

## :host-content

## Rating组件

主要为了帮大家演示下如何实现组件那节课布置的作业。

参照目标： <https://valor-software.com/ngx-bootstrap/#/rating>

步骤：

1、如何显示一个星星。

2、如何显示一个空心的星星。

3、如何显示定义数量个的星星。

4、如何让定义数量个的星星有的空心有的实心。

5、如何把评分数值传递到组件中，根据评分数值显示个数。

6、让传递进来的值控制最大的可显示的星星数。

7、添加点击事件来操作评分，并改变对于星星个数。

8、当可点击之后，我们通过变量来控制是否能点击。

9、把点击后的星星发射出去。

7+8 就是双向数据绑定的分解。

# 作业

自学Angular动画，包括路由动画

官网地址：<https://www.angular.cn/guide/animations>