## 指令笔记

ngIf  
 如果你希望根据一个条件来决定显示或隐藏一个元素，可以使用ngIf指令。这个条件是由你  
 传给指令的表达式的结果决定的。  
 如果表达式的结果返回的是一个假值，那么元素会从DOM上被移除。  
  
ngSwitch  
 有时候你需要根据一个给定的条件来渲染不同的元素。  
 遇到这种情况  
 <div class="container">  
 <div \*ngIf="myVar == 'A'">Var is A</div>  
 <div \*ngIf="myVar == 'B'">Var is B</div>  
 <div \*ngIf="myVar == 'C'">Var is C</div>  
 <div \*ngIf="myVar != 'A' && myVar != 'B' && myVar != 'C'">Var is something else</div>  
 </div>  
 指令背后的思想也是一样的：对表达式进行一次求值，然后根据其结果来决定如何显示指令内的嵌套元素。  
 一旦有了结果，我们就可以：  
  使用ngSwitchCase指令描述已知结果；  
  使用ngSwitchDefault指令处理所有其他未知情况  
 <div class="container" [ngSwitch]="myVar">  
 <div \*ngSwitchCase="'A'">Var is A</div>  
 <div \*ngSwitchCase="'B'">Var is B</div>  
 <div \*ngSwitchDefault>Var is something else</div>  
 </div>  
  
 不需要修改默认（即备用）条件。  
 ngSwitchDefault元素是可选的。如果我们不用它，那么当myVar没有匹配到任何期望的值时就不会渲染任何东西。  
 你也可以为不同的元素声明同样的\*ngSwitchCase值，这样就可以多次匹配同一个值了  
  
ngStyle  
 用ngStyle指令，可以通过Angular表达式给特定的DOM元素设定CSS属性。  
 该指令最简单的用法就是[style.<cssproperty>]="value"的形式  
 <div [style.background-color]="'yellow'">  
 Uses fixed yellow background  
 </div>  
 这个代码片段就是使用ngStyle指令把CSS的background-color属性设置为字符串字面量  
 yellow。  
  
 另一种设置固定值的方式就是使用ngStyle属性，使用键值对来设置每个属性。  
 <div [ngStyle]="{color: 'white', 'background-color': 'blue'}">  
 Uses fixed white text on blue background  
 </div>  
 注意，在ngStyle的说明中，我们对background-color使用了单引号，但却没  
 有对color使用。这是为什么呢？因为ngStyle的参数是一个JavaScript对象，而  
 color是一个合法的键，不需要引号。但是在background-color中，连字符是  
 不允许出现在对象的键名当中的，除非它是一个字符串，因此使用了引号。  
 通常情况下，我尽量不会对对象的键使用引号，除非不得不用。   
  
 动态例子：  
 <div class="ui input">  
 <input type="text" value="{{color}}" #colorinput placeholder="请输入color">  
 </div>  
 <div class="ui input">  
 <input type="text" value="{{fontSize}}" #fontinput placeholder="请输入fontsize">  
 </div>  
 <button class="ui primary button" (click)="apply(colorinput.value, fontinput.value)">  
 Apply settings  
 </button>  
 <div>  
 <span [ngStyle]="{color: 'red'}" [style.font-size.px]="fontSize">  
 red text  
 </span>  
 </div>  
 <hr>  
 <h4 class="ui horizontal divider header">  
 ngStyle with object property from variable  
 </h4>  
 <div>  
 <span [ngStyle]="{color: color}">  
 {{ color }} text  
 </span>  
 </div>  
  
 <hr>  
 <h4 class="ui horizontal divider header">  
 style from variable  
 </h4>  
 <div [style.background-color]="color"  
 style="color: white;">  
 {{ color }} background  
 </div>  
   
 apply(color: string, fontSize: number) {  
 this.color = color;  
 this.fontSize = fontSize;  
 }

效果如下

