03月1日 日报

* 已完成的内容

在上面的代码中，我们将ElementRef和Renderer2注入构造函数，前者会得到添加了appButton属性的元素，后者是Angular提供的DOM API操作。 注：ElementRef和Renderer2后续会详解。 然后在demo-directive.component.html中添加：

<button appButton>指令按钮/button>

当你打开网页时，会发现此button添加上了btn样式类(背景色变蓝，字体颜色变白色)。 当然，还有一些快捷的方式来给使用了指令的宿主元素添加属性和事件： (1) @HostBinding() 再来修改一下button.directive.ts：

export class ButtonDirective {

@HostBinding('style.font-size') fontSize = '20px';

...

}

当你添加了上面的代码时，你会发现demo-directive.component.html中的按钮字体变大了。 注：@HostBinding()里是的属性和直接属性绑定是一样的效果，主要给宿主元素设置属性。 (2) @HostListener 我们还可以为宿主元素绑定事件：

// button.directive.ts

@HostListener('click') onClick() {

alert('你点了我!');

}

添加如上代码后，你可以试试点击按钮。 注：@HostListener()里是事件名称，后面跟着一个监听函数，名称可以是任意合法的字符。 有些时候我们需要给这个指令传递一些值，如何实现呢？ 还记得前面《组件通讯(@Input和@Output)》一章中我们讲过@Input()可以设置别名。 修改button.directive.ts：

export class ButtonDirective implements AfterViewInit {

...

@Input('appButton') name: string; // 定义别名

constructor(private er: ElementRef, private renderer2: Renderer2) { }

ngAfterViewInit() {

...

if (this.name) {

const text = this.renderer2.createText(this.name);

this.renderer2.appendChild(this.er.nativeElement, text);

}

}

}

通过@Input()定义与指令同名的输入属性，我们就可以给指令传递参数了：

// demo-directive.component.html

<button appButton="额外名称">指令按钮</button>

结构型指令 最常用的内置结构型指令有ngIf, ngFor, ngSwitch。 (1) ngIf 在demo-directive.component.html中添加：

<div>

<button (click)="isShow = !isShow">点击试试</button>

<div \*ngIf="isShow">ngIf结构型指令</div>

</div>

随着点击按钮，isShow属性值会在true和false之间切换，而div也会跟着显示或隐藏，其实是插入或移除。 而在Angular4中，新增else效果(使用模板变量#name)：

<button (click)="isShowElse = !isShowElse">点击试试ngIf else</button>

<div \*ngIf="isShowElse else next">ngIf结构型指令中的else</div>

<ng-template #next>else效果</ng-template>

遇到的问题及解决方案

* 教师评语：