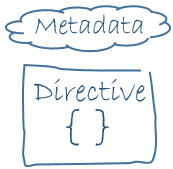
**Directives**



Angular templates are *dynamic*. When Angular renders them, it transforms the DOM according to the instructions given by **directives**.

A directive is a class with a @Directive decorator. A component is a *directive-with-a-template*; a @Component decorator is actually a @Directive decorator extended with template-oriented features. 

While **a component is technically a directive**, components are so distinctive and central to Angular applications that this architectural overview separates components from directives.

Two *other* kinds of directives exist: *structural* and *attribute* directives.

They tend to appear within an element tag as attributes do, sometimes by name but more often as the target of an assignment or a binding.

**Structural** directives alter layout by adding, removing, and replacing elements in DOM.

The [example template](https://angular.io/docs/ts/latest/guide/architecture.html#templates) uses two built-in structural directives:

**app/hero-list.component.html (structural)**

<li \*ngFor="let hero of heroes"></li>

<hero-detail \*ngIf="selectedHero"></hero-detail>

* [\*ngFor](https://angular.io/docs/ts/latest/guide/displaying-data.html#ngFor) tells Angular to stamp out one <li> per hero in the heroes list.
* [\*ngIf](https://angular.io/docs/ts/latest/guide/displaying-data.html#ngIf) includes the HeroDetail component only if a selected hero exists.

**Attribute** directives alter the appearance or behavior of an existing element. In templates they look like regular HTML attributes, hence the name.

The ngModel directive, which implements two-way data binding, is an example of an attribute directive. ngModel modifies the behavior of an existing element (typically an <input>) by setting its display value property and responding to change events.

**app/hero-detail.component.html (ngModel)**

<input [(ngModel)]="hero.name">

Angular has a few more directives that either alter the layout structure (for example, [ngSwitch](https://angular.io/docs/ts/latest/guide/template-syntax.html" \l "ngSwitch)) or modify aspects of DOM elements and components (for example, [ngStyle](https://angular.io/docs/ts/latest/guide/template-syntax.html" \l "ngStyle) and [ngClass](https://angular.io/docs/ts/latest/guide/template-syntax.html" \l "ngClass)).

Of course, you can also write your own directives. Components such as HeroListComponent are one kind of custom directive.