

Template Basics

Bindings/interpolation

Your favorite hero is: {{vm.favoriteHero}}

Bindings/interpolation

Your favorite hero is: {{vm.favoriteHero}}

Filters

<td>{{movie.title | uppercase}}</td>

Pipes

<td>{{movie.title | uppercase}}</td>

Local variables

```
<tr ng-repeat="movie in vm.movies">
<td>{{movie.title}}</td>
</tr>
```

Input variables

```
<tr *ngFor="let movie of movies">
<td>{{movie.title}}</td>
</tr>
```

Template directives

ng-class

```
<div ng-class="{active: isActive}">
<div ng-class="{active: isActive, shazam: isImportant}">
```

ngClass

```
<div [ngClass]="{'active': isActive}">
<div [ngClass]="{'active': isActive, 'shazam': isImportant}">
<div [class.active]="isActive">
```

ng-click

```
<button ng-click="vm.toggleImage()">
<button ng-click="vm.toggleImage($event)">
```

Bind to click event

```
<button (click)="toggleImage()">
<button (click)="toggleImage($event)">
```

ng-controller

```
<div ng-controller="MovieListCtrl as vm">
```

Component Decorator

```
@Component({
  selector: 'app-movie-list',
  templateUrl: './movie-list.component.html',
  styleUrls: ['./movie-list.component.css'],
})
```

ng-href

```
<a ng-href="{{ angularDocsUrl }}">Angular Docs</a>
```

Bind to href property

```
<a [href]="angularDocsUrl">Angular Docs</a>
```

ng-if

```
<table ng-if="movies.length">
```

*ngIf

```
<table *ngIf="movies.length">
```

ng-model

```
<input ng-model="vm.favoriteHero"/>
```

ngModel

```
<input [(ngModel)]="favoriteHero" />
```

ng-repeat

```
<tr ng-repeat="movie in vm.movies">
```

*ngFor

```
<tr *ngFor="let movie of movies">
```

ng-show

```
<h3 ng-show="vm.favoriteHero">
Your favorite hero is: {{vm.favoriteHero}}
</h3>
```

Bind to hidden property

```
<h3 [hidden]="!favoriteHero">
Your favorite hero is: {{favoriteHero}}
</h3>
```

ng-src

```

```

Bind to src property

```
<img [src]="movie.imageUrl">
```

ng-style

```
<div ng-style="{color: colorPreference}">
```

ngStyle

```
<div [ngStyle]="{'color': colorPreference}">
<div [style.color]="colorPreference">
```

Template directives (cont)

ng-switch

```
<div ng-switch="vm.favoriteHero &&
vm.checkMovieHero(vm.favoriteHero)">
<div ng-switch-when="true">
Excellent choice!
</div>
<div ng-switch-when="false">
No movie, sorry!
</div>
<div ng-switch-default>
Please enter your favorite hero.
</div>
</div>
```

[ngSwitch]

```
<span [ngSwitch]="favoriteHero &&
checkMovieHero(favoriteHero)">
<p ngSwitchCase="true">
Excellent choice!
</p>
<p ngSwitchCase="false">
No movie, sorry!
</p>
<p *ngSwitchDefault>
Please enter your favorite hero.
</p>
</span>
```

Filters/pipes

Currency

```
<td>{{movie.price | currency}}</td>
```

Currency

```
<td>{{movie.price | currency:'USD':true}}</td>
```

Date

```
<td>{{movie.releaseDate | date}}</td>
```

Date

```
<td>{{movie.releaseDate | date}}</td>
```

Filter

```
<tr ng-repeat="movie in movieList | filter: {title:listFilter}">
```

None

If you need the same filtering code in several templates, consider building a custom pipe.

json

```
<pre>{{movie | json}}</pre>
```

json

```
<pre>{{movie | json}}</pre>
```

Limit to

```
<tr ng-repeat="movie in movieList | limitTo:2:0">
```

Slice

```
<tr *ngFor="let movie of movies | slice:0:2">
```

LowerCase

```
<td>{{movie.title | lowercase}}</td>
```

LowerCase

```
<td>{{movie.title | lowercase}}</td>
```

Number

```
<td>{{movie.starRating | number}}</td>
```

Number

```
<td>{{movie.starRating | number}}</td>
<td>{{movie.starRating | number:'1.1-2'}}</td>
<td>{{movie.approvalRating | percent: '1.0-2'}}</td>
```

OrderBy

```
<tr ng-repeat="movie in movieList | orderBy : 'title'">
```

None

If you need the same ordering or sorting code in several templates, consider building a custom pipe.



Modules/controllers/components	
IIFE	None
<pre>(function () { ... })();</pre>	This is a nonissue in Angular because ES 2015 modules handle the namespacing for you.
Angular Modules	NgModules
<pre>angular.module("movieHunter", ["ngRoute"]);</pre>	<pre>import { NgModule } from '@angular/core'; import { BrowserModule } from '@angular/platform-browser'; import { AppComponent } from './app.component'; @NgModule({ imports: [BrowserModule], declarations: [AppComponent], bootstrap: [AppComponent] }) export class AppModule { }</pre>
Controller Registration	Component Decorator
<pre>angular .module("movieHunter") .controller("MovieListCtrl", ["movieService", MovieListCtrl]);</pre>	<pre>@Component({ selector: 'app-movie-list', templateUrl: './movie-list.component.html', styleUrls: ['./movie-list.component.css'], })</pre>
Controller Function	Component Class
<pre>function MovieListCtrl(movieService) { }</pre>	<pre>export class MovieListComponent { }</pre>
Dependency Injection	Dependency Injection
<pre>MovieListCtrl.\$inject = ['MovieService']; function MovieListCtrl(movieService) { }</pre>	<pre>constructor(movieService: MovieService) { }</pre>

Styles Sheets

Link Tag

```
<link href="styles.css" rel="stylesheet" />
```

Style Configuration

```
"styles": [  
  "styles.css"  
],
```

Style Urls

```
styleUrls: [ './movie-list.component.css' ],
```

C

By **Jeanvaljean** (Mobbigdick)
cheatography.com/mobbigdick/

Not published yet.
Last updated 7th September, 2018.
Page 4 of 4.

Sponsored by **Readability-Score.com**
Measure your website readability!
<https://readability-score.com>