1. Let's add two properties to our AppComponent,

a title property for the application name

and a hero property for a hero named "Windstorm".

**import** {Component} **from '@angular/core'**;  
  
@Component({  
 **selector**: **'my-app'**,  
 **template**: **'<h1>{{title}}</h1><h2>{{hero}} details!</h2>'**})  
**export class** AppComponent {  
 **title** = **"Tour of Heroes"**;  
 **hero** = **'Windstorm'**;  
}

The double curly braces tell our app to read the title and hero properties from the component and render them. This is the "interpolation" form of one-way data binding.

Learn more about interpolation in the [Displaying Data chapter](https://angular.io/docs/ts/latest/guide/displaying-data.html).

1. **Converting Hero to an object**

**建立一个新的类Hero,目前只有两个属性 id 和 name**

**export class** Hero{  
 **id**:**number**;  
 **name**:**string**;  
}

注意要使用这个新建的类，首先需要导入

**import** {Component} **from '@angular/core'**;  
**import** {Hero} **from './hero'**;  
  
@Component({  
 **selector**: **'my-app'**,  
 **template**: **'<h1>{{title}}</h1><h2>{{hero.name}} details!</h2>'**})  
**export class** AppComponent {  
 **title** = **'Tour of Heroes'**;  
 **hero**:Hero = {  
 **id**:1,  
 **name** : **'Windstorm'** } // 初始化这个hero Object

}

复杂一点的模板

**import** {Component} **from '@angular/core'**;  
**import** {Hero} **from './hero'**;  
  
@Component({  
 **selector**: **'my-app'**,  
 **template**:**`  
 <h1>{{title}}</h1>  
 <h2>{{hero.name}} details!</h2>  
 <div><label>id: </label>{{hero.id}}</div>  
 <div><label>name: </label>{{hero.name}}</div>  
 `**})  
**export class** AppComponent {  
 **title** = **'Tour of Heroes'**;  
 **hero**:Hero = {  
 **id**:1,  
 **name** : **'Windstorm'** }  
}

当然也可以使用模板文件

**app/tmpl/app.component.html**

<**h1**>{{**title**}}</**h1**>  
<**h2**>{{**hero**.**name**}} details!</**h2**>  
<**div**><**label**>id: </**label**>{{**hero**.**id**}}</**div**>  
<**div**><**label**>name: </**label**>{{**hero**.**name**}}</**div**>

**import** {Component} **from '@angular/core'**;  
**import** {Hero} **from './hero'**;  
  
@Component({  
 **selector**: **'my-app'**,  
 **templateUrl**: **'app/tmpl/app.component.html'**})  
**export class** AppComponent {  
 **title** = **'Tour of Heroes'**;  
 **hero**:Hero = {  
 **id**:1,  
 **name** : **'Windstorm'** }  
}

改变模板内容，使得name属性可编辑

<**h1**>{{**title**}}</**h1**>  
<**h2**>{{**hero**.**name**}} details!</**h2**>  
<**div**><**label**>id: </**label**>{{**hero**.**id**}}</**div**>  
<**div**>  
 <**label**>name: </**label**>  
 <**input value="{{hero.name}}" placeholder="name"**>  
</**div**>

**注意此时只是one way binding**

下面实现2 way binding

1. **Two-Way Binding**

We intend to display the name of the hero in the <input>, change it, and see those changes wherever we bind to the hero’s name. In short, we want two-way data binding.

Before we can use two-way data binding for **form inputs**, **we need to import the FormsModule package in our Angular module.**

**We add it to the NgModule decorator's imports array. This array contains the list of external modules used by our application.**

Now we have included the forms package which includes ngModel.

**import** { NgModule } **from '@angular/core'**;  
**import** { BrowserModule } **from '@angular/platform-browser'**;  
  
**import** { FormsModule } **from '@angular/forms'**; *// for 2 way binding***import** { AppComponent } **from './app.component'**;  
  
@NgModule({  
 **imports**: [ BrowserModule, FormsModule ],  
 **declarations**: [ AppComponent ],  
 **bootstrap**: [ AppComponent ]  
})  
**export class** AppModule { }

Learn more about the FormsModule and ngModel in the [Forms](https://angular.io/docs/ts/latest/guide/forms.html#ngModel) and [Template Syntax](https://angular.io/docs/ts/latest/guide/template-syntax.html#ngModel) chapters.

Let’s update the template to use the **ngModel** built-in directive for two-way binding.

<**h1**>{{**title**}}</**h1**>  
<**h2**>{{**hero**.**name**}} details!</**h2**>  
<**div**><**label**>id: </**label**>{{**hero**.**id**}}</**div**>  
<**div**>  
 <**label**>name(1 way binding): </**label**>  
 <**input type="text" value="{{ hero.name }}" placeholder="name"** /><**br**/>  
 <**label**>name(2 way binding): </**label**>  
 <**input [(ngModel)]="hero.name" placeholder="name"**>  
</**div**>

注意不需要在app.component.ts 当中再次导入

下一章内容

## The Road Ahead

Our Tour of Heroes only displays one hero and we really want to display a list of heroes. We also want to allow the user to select a hero and display their details. We’ll learn more about how to retrieve lists, bind them to the template, and allow a user to select a hero in the [next tutorial chapter](https://angular.io/docs/ts/latest/tutorial/toh-pt2.html).

1. 首先建立一个hero.service.ts文件，用于模拟获得hero的data 列表。记得导入Hero对象
2. **import** {Hero} **from "./hero"**;  
     
   **const** HEROES: Hero[] = [  
    { **id**: 11, **name**: **'Mr. Nice'** },  
    { **id**: 12, **name**: **'Narco'** },  
    { **id**: 13, **name**: **'Bombasto'** },  
    { **id**: 14, **name**: **'Celeritas'** },  
    { **id**: 15, **name**: **'Magneta'** },  
    { **id**: 16, **name**: **'RubberMan'** },  
    { **id**: 17, **name**: **'Dynama'** },  
    { **id**: 18, **name**: **'Dr IQ'** },  
    { **id**: 19, **name**: **'Magma'** },  
    { **id**: 20, **name**: **'Tornado'** }  
   ];  
     
   **export class** Heroes{  
    **static** *HEROES*:Hero[] = HEROES;  
   }

**一定要记得export, 不然不是public 的属性**

1. 在AppComponent 当中进行导入
2. **import** {Component} **from '@angular/core'**;  
   **import** {Hero} **from './hero'**;  
   **import** {Heroes} **from './hero.service'**;  
     
   @Component({  
    **selector**: **'my-app'**,  
    **templateUrl**: **'app/tmpl/app.component.html'**})  
   **export class** AppComponent {  
    **title** = **'Tour of Heroes'**;  
    *// hero:Hero = {  
    // id:1,  
    // name : 'Windstorm'  
    // }* **heroes**:Hero[] = Heroes.*HEROES*;  
     
    **constructor**(){  
    *//this.heroes = Heroes.HEROES;* ***console***.log(**this**.**heroes**);  
    }  
   }

然后在模板当中进行显示

<**h1**>{{**title**}}</**h1**>  
<**div \*ngFor="let hero of heroes"**>  
 <**h2**>{{**hero**.*name*}} details!</**h2**>  
 <**div**><**label**>id: </**label**>{{**hero**.*id*}}</**div**>  
 <**div**>  
 <**label**>name(1 way binding): </**label**>  
 <**input type="text" value="{{ hero.***name* **}}" placeholder="name"** /><**br**/>  
 <**label**>name(2 way binding): </**label**>  
 <**input [(ngModel)]="hero.***name***" placeholder="name"**>  
 </**div**>  
</**div**>

**注意#hero of heroes 这种写法已经过时了**

The leading asterisk (\*) in front of ngFor is a critical part of this syntax.

The (\*) prefix to ngFor indicates that the <li> element and its children constitute a master template.

The ngFor directive iterates over the heroes array returned by the AppComponent.heroes property and stamps out instances of this template.

The quoted text assigned to ngFor means “take each hero in the*heroes*array, store it in the local*hero*variable, and make it available to the corresponding template instance”.

The let keyword before "hero" identifies hero as a template input variable. We can reference this variable within the template to access a hero’s properties.

Learn more about ngFor and template input variables in the [Displaying Data](https://angular.io/docs/ts/latest/guide/displaying-data.html#ngFor) and [Template Syntax](https://angular.io/docs/ts/latest/guide/template-syntax.html#ngFor) chapters.

换成简单的列表模板

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

<span class="badge">{{hero.id}}</span> {{hero.name}}

</li>

### Styling our heroes

教程当中是直接写在AppComponent.ts的@component decorator 当中的。

**import** {Component} **from '@angular/core'**;  
**import** {Hero} **from './hero'**;  
**import** {Heroes} **from './hero.service'**;  
  
@Component({  
 **selector**: **'my-app'**,  
 **templateUrl**: **'app/tmpl/app.component.html'**,  
 **styles**: [**`  
 .selected {  
 background-color: #CFD8DC !important;  
 color: white;  
 }  
 .heroes {  
 margin: 0 0 2em 0;  
 list-style-type: none;  
 padding: 0;  
 width: 15em;  
 }  
 .heroes li {  
 cursor: pointer;  
 position: relative;  
 left: 0;  
 background-color: #EEE;  
 margin: .5em;  
 padding: .3em 0;  
 height: 1.6em;  
 border-radius: 4px;  
 }  
 .heroes li.selected:hover {  
 background-color: #BBD8DC !important;  
 color: white;  
 }  
 .heroes li:hover {  
 color: #607D8B;  
 background-color: #DDD;  
 left: .1em;  
 }  
 .heroes .text {  
 position: relative;  
 top: -3px;  
 }  
 .heroes .badge {  
 display: inline-block;  
 font-size: small;  
 color: white;  
 padding: 0.8em 0.7em 0 0.7em;  
 background-color: #607D8B;  
 line-height: 1em;  
 position: relative;  
 left: -1px;  
 top: -4px;  
 height: 1.8em;  
 margin-right: .8em;  
 border-radius: 4px 0 0 4px;  
 }  
`**]  
})  
**export class** AppComponent {  
 **title** = **'Tour of Heroes'**;  
 *// hero:Hero = {  
 // id:1,  
 // name : 'Windstorm'  
 // }* **heroes**: Hero[] = Heroes.*HEROES*;  
  
 **constructor**() {  
 *//this.heroes = Heroes.HEROES;* ***console***.log(**this**.**heroes**);  
 }  
}

然后模板修改为

<**h2**>My Heroes</**h2**>  
<**ul class="heroes"**>  
 <**li \*ngFor="let hero of heroes"**>  
 <**span class="badge"**>{{**hero**.*id*}}</**span**> {{**hero**.*name*}}  
 </**li**>  
</**ul**>



实际上最好事单独将css 放入一个文件

**app/css/app.component.css**

.**selected** {  
 **background-color**: **#CFD8DC !important**;  
 **color**: **white**;  
}  
.**heroes** {  
 **margin**: 0 0 2**em** 0;  
 **list-style-type**: **none**;  
 **padding**: 0;  
 **width**: 15**em**;  
}  
.**heroes li** {  
 **cursor**: **pointer**;  
 **position**: **relative**;  
 **left**: 0;  
 **background-color**: **#EEE**;  
 **margin**: .5**em**;  
 **padding**: .3**em** 0;  
 **height**: 1.6**em**;  
 **border-radius**: 4**px**;  
}  
.**heroes li**.**selected**:**hover** {  
 **background-color**: **#BBD8DC !important**;  
 **color**: **white**;  
}  
.**heroes li**:**hover** {  
 **color**: **#607D8B**;  
 **background-color**: **#DDD**;  
 **left**: .1**em**;  
}  
.**heroes** .**text** {  
 **position**: **relative**;  
 **top**: -3**px**;  
}  
.**heroes** .**badge** {  
 **display**: **inline-block**;  
 **font-size**: **small**;  
 **color**: **white**;  
 **padding**: 0.8**em** 0.7**em** 0 0.7**em**;  
 **background-color**: **#607D8B**;  
 **line-height**: 1**em**;  
 **position**: **relative**;  
 **left**: -1**px**;  
 **top**: -4**px**;  
 **height**: 1.8**em**;  
 **margin-right**: .8**em**;  
 **border-radius**: 4**px** 0 0 4**px**;  
}

**import** {Component} **from '@angular/core'**;  
**import** {Hero} **from './hero'**;  
**import** {Heroes} **from './hero.service'**;  
  
@Component({  
 **selector**: **'my-app'**,  
 **templateUrl**: **'app/tmpl/app.component.html'**,  
 **styleUrls**: [**'app/css/app.component.css'**]  
})  
**export class** AppComponent {  
 **title** = **'Tour of Heroes'**;  
 *// hero:Hero = {  
 // id:1,  
 // name : 'Windstorm'  
 // }* **heroes**: Hero[] = Heroes.*HEROES*;  
  
 **constructor**() {  
 *//this.heroes = Heroes.HEROES;* ***console***.log(**this**.**heroes**);  
 }  
}