* {{ hero.name }} – {{class property}}
* <h2>{{ hero.name | uppercase }} Details</h2> - piping value through some processing

The word [uppercase](https://angular.io/api/common/UpperCasePipe) in the interpolation binding, right after the pipe operator ( | ), activates the built-in UppercasePipe.

[Pipes](https://angular.io/guide/pipes) are a good way to format strings, currency amounts, dates and other display data. Angular ships with several built-in pipes and you can create your own.

---------------------------------------------------------------------------------------------------------

<div>

<label>name:

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

</label>

</div>

**[(ngModel)]** is Angular's two-way data binding syntax.

Here it binds the hero.name property to the HTML textbox so that data can flow in both directions: from the hero.name property to the textbox, and from the textbox back to the hero.name.

imports: [

BrowserModule,

FormsModule

],

declarations: [

AppComponent,

HeroesComponent

],

<div **\*ngIf**="selectedHero">

<h2>{{ selectedHero.name | uppercase }} Details</h2>

<div><span>id: </span>{{selectedHero.id}}</div>

<div>

<label>name:

<input **[(ngModel)]="selectedHero.name"** placeholder="name">

</label>

</div>

</div>

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

[class.selected]="hero === selectedHero"

**(click)="onSelect(hero)"**>

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

</li>

The hero property [must be an *Input* property](https://angular.io/guide/template-syntax#inputs-outputs), annotated with the @[Input](https://angular.io/api/core/Input)() decorator, because the *external* HeroesComponent [will bind to it](https://angular.io/tutorial/toh-pt3#heroes-component-template) like this.

<app-hero-detail [hero]="selectedHero"></app-hero-detail>

Amend the @angular/core import statement to include the [Input](https://angular.io/api/core/Input) symbol.

src/app/hero-detail/hero-detail.component.ts (import Input)

import { [Component](https://angular.io/api/core/Component), [OnInit](https://angular.io/api/core/OnInit), [Input](https://angular.io/api/core/Input) } from '@angular/core';

Add a hero property, preceded by the @[Input](https://angular.io/api/core/Input)() decorator.

@[Input](https://angular.io/api/core/Input)() hero: Hero;

[hero]="selectedHero" is an Angular [property binding](https://angular.io/guide/template-syntax#property-binding).

It's a one way data binding from the selectedHero property of the HeroesComponent to the hero property of the target element, which maps to the hero property of the HeroDetailComponent.

Now when the user clicks a hero in the list, the selectedHero changes. When the selectedHero changes, the property binding updates hero and the HeroDetailComponent displays the new hero.

--------------------------------------------------------------------------------------------------------