



Angular : Directive & Pipe

- ☐ Able to use Directive

- ☐ Able to use Pipes



- ❑ There are three kinds of directives in Angular:
 - ❑ Components—directives with a template.
 - ❑ Structural directives—change the DOM layout by adding and removing DOM elements.
 - ❑ Attribute directives—change the appearance or behavior of an element, component, or another directive.

❑ *ngIf and *ngFor

- ❑ Prefix with an astrick
- ❑ Assign to a quoted string expression

Structural Directives

*ngIf: If logic

*ngFor: For loops

*ngIf Built-In Directive

```
restaurant-list.component.ts  restaurant-list.component.html x
1  <div class="panel panel-primary">
2    <div class="panel-heading">
3      {{pageTitle}}
4    </div>
5  <div class="panel-body">
6    <div class="row">
7      <div class="col-md-2"> filter by: </div>
8      <div class="col-md-4">
9        <input type="text"/>
10      </div>
11    </div>
12    <div class="table-responsive">
13      <table class="table">
14        | *ngIf='restaurants && restaurants.length' >
15        <thead>
16          <tr>
17            <th>
18              <button class="btn btn-primary">
19                Show Image
20              </button>
21            </th>
22            <th> Title </th>
23            <th> City </th>
24            <th> State </th>
25            <th> 5 Star Rank </th>
26          </tr>
27        </thead>
28        <tbody>
29        </tbody>
30      </table>
31    </div>
32  </div>
```

❑ *ngIf

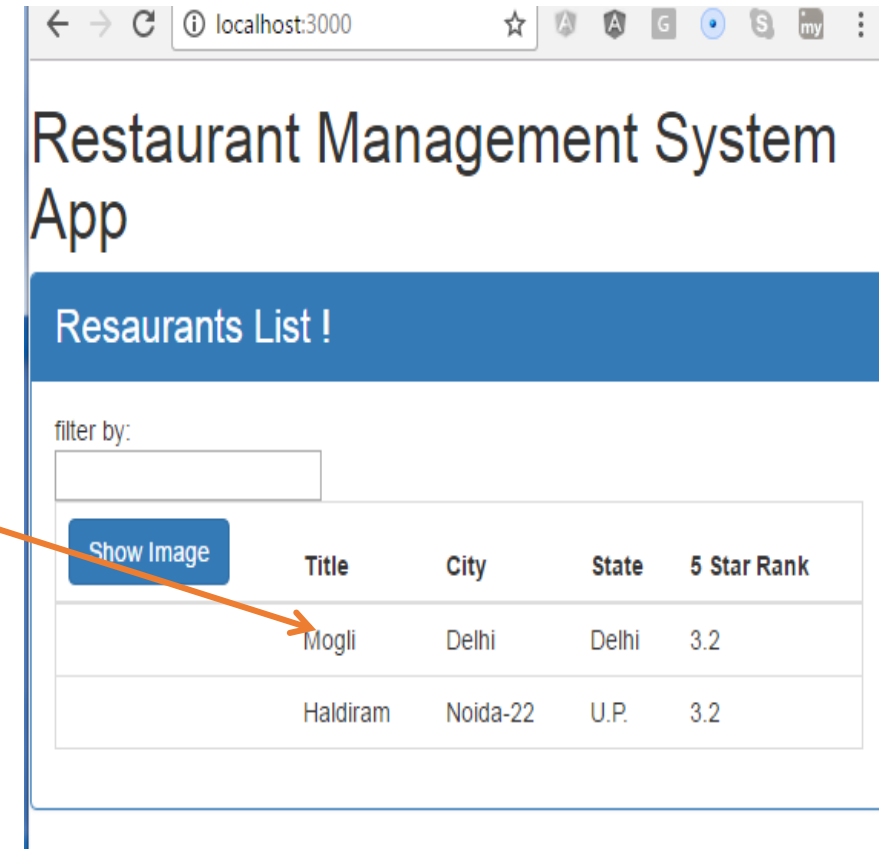
- ❑ Expression is evaluated as a true or false value .
- ❑ Conditionally includes a template based on the value of an expression.

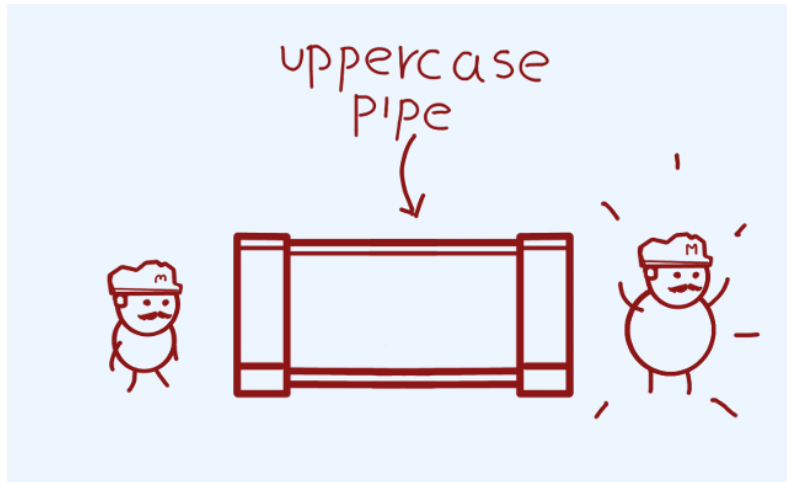
*ngForBuilt-In Directive

❑ *ngFor

- ❑ Define the local variable with let
- ❑ Specify ' of': 'let product of restaurants'

```
restaurant-list-component.html x
<tr>
  <th>
    <button class="btn btn-primary">
      Show Image
    </button>
  </th>
  <th> Title </th>
  <th> City </th>
  <th> State </th>
  <th> 5 Star Rank </th>
</tr>
</thead>
<tbody>
  <tr *ngFor='let restaurant of restaurants'>
    <td></td>
    <td>{{ restaurant.restaurantTitle }}
    <td>{{ restaurant.restaurantCity }}
    <td>{{ restaurant.restaurantState }}
    <td>{{ restaurant.starRating }}
  </tr>
</tbody>
</table>
</div>
</div>
```





- ❑ Transform bound properties before display
- ❑ Built-in pipes
 - ❑ Date
 - ❑ number, decimal, percent, currency
 - ❑ json, slice
 - ❑ Etc.
 - ❑ Example
 - ❑ `{{ resaurant.restaurantTitle | lowercase }}`
- ❑ Pipe parameters
 - ❑ Separated with colons
- ❑ Custom pipes

P AsyncPipe

P DecimalPipe

P DeprecatedDecimalPipe

P I18nSelectPipe

P LowerCasePipe

P TitleCasePipe

P CurrencyPipe

P DeprecatedCurrencyPipe

P DeprecatedPercentPipe

P JsonPipe

P PercentPipe

P UpperCasePipe

P DatePipe

P DeprecatedDatePipe

P I18nPluralPipe

P KeyValuePipe

P SlicePipe


```
import { Pipe, PipeTransform } from '@angular/core';
```

```
@Pipe({
```

```
  name : 'restaurantFilter'
```

```
})
```

```
export class RestaurantFilterPipe
```

```
  implements PipeTransform{
```

```
    transform (value : IRestaurant [],
```

```
              filterBy: string) : IRestaurant []{
```

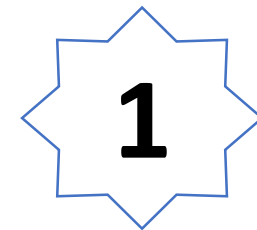
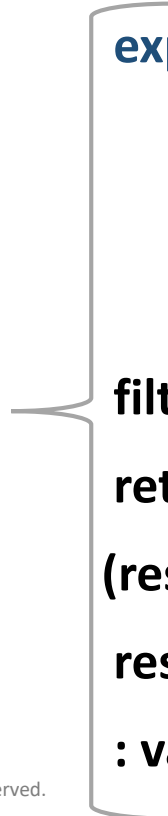
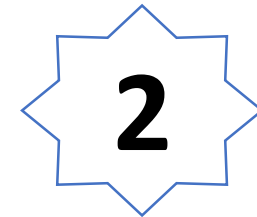
```
    filterBy= (filterBy) ? filterBy.toLocaleLowerCase() : null;
```

```
    return (filterBy) ? value.filter(
```

```
      (restaurant : IRestaurant) =>
```

```
        restaurant.restaurantTitle.toLocaleLowerCase().indexOf(filterBy) !== -1)
```

```
      : value;    } }
```



Template

```
<tr *ngFor='let restaurant of restaurants | restaurantFilter : listFilter'>  
  <td>
```

Module

```
app.component.ts  app.module.ts x  restaurant-filter.pipes.ts  restaurant-list-component.html  
1  import { NgModule } from '@angular/core';  
2  import { BrowserModule } from '@angular/platform-browser';  
3  import { FormsModule } from '@angular/forms';  
4  import { AppComponent } from './app.component';  
5  import { RestaurantFilterPipe } from './restaurants/restaurant-filter.pipes';  
6  import { RestaurantListComponent } from './restaurants/restaurant-list.component';  
7  @NgModule({  
8    imports: [ BrowserModule, FormsModule ],  
9    declarations: [ AppComponent, RestaurantListComponent, RestaurantFilterPipe ],  
10   bootstrap: [ AppComponent ]  
11 })  
12 export class AppModule { }  
13
```

- Pipes are pure by default
- You make a pipe impure by setting its pure flag to false.

```
@Pipe(  
  { name: 'restaurantFilter',  
    pure: false  
  })
```

Pure pipes –

- Angular executes a pure pipe only when it detects a pure change to the input value.

Impure pipes –

- Angular executes an impure pipe during every component change detection cycle. An impure pipe is called often, as often as every keystroke or mouse-move.
- With that concern in mind, implement an impure pipe with great care. An expensive, long-running pipe could destroy the user experience.



Infogain Corporation, HQ

485 Alberto Way Los Gatos,
CA 95032 USA
Phone: 408-355-6000
Fax: 408-355-7000

Pune

7th Floor, Bhalerao Towers, CTS No.1669 -
1670, Behind Hotel Pride,
Shivaji Nagar, Pune - 411005
Phone : +91-20-66236700

Infogain Irvine

41 Corporate Park,
Suite 390 Irvine, CA 2606 USA
Phone: 949-223-5100
Fax: 949-223-5110

Infogain Austin

Stratum Executive Center Building D
11044 Research Boulevard Suite 200
Austin, Texas 78759

Noida

A-16, Sector 60, Noida Gautam Budh agar,
201301 (U.P.) India
Phone: +91-120-2445144
Fax: +91-120-2580406

Dubai

P O Box 500588 Office No.105,
Building No. 4, Dubai Outsource Zone,
Dubai, United Arab Emirates
Tel: +971-4-458-7336